

Carla Indira Fernandes Gonçalves

# Desenvolvimento de aplicação orientada a objecto

“O caso da Delegação do Ministério da Educação e Desporto do concelho da Praia”

**Universidade Jean Piaget de Cabo Verde**

**Campus Universitário da Cidade da Praia  
Caixa Postal 775, Palmarejo Grande  
Cidade da Praia, Santiago  
Cabo Verde**

**26.12.11**



Carla Indira Fernandes Gonçalves

# Desenvolvimento de aplicação orientada a objecto

“O caso da Delegação do Ministério da Educação e Desporto do concelho da Praia”

**Universidade Jean Piaget de Cabo Verde**

**Campus Universitário da Cidade da Praia  
Caixa Postal 775, Palmarejo Grande  
Cidade da Praia, Santiago  
Cabo Verde**

**26.12.11**

Eu, Carla Indira Fernandes Gonçalves, autora da monografia intitulada *Desenvolvimento de aplicação orientada a objecto: o caso da Delegação do Ministério da Educação e Desporto do Concelho da Praia*, declaro que, salvo fontes devidamente citadas e referidas, o presente documento é fruto do meu trabalho pessoal, individual e original.

Cidade da Praia aos 26 dias do Dezembro de 2011

Carla Indira Fernandes Gonçalves

Memória Monográfica apresentada à Universidade Jean Piaget de Cabo Verde como parte dos requisitos para a obtenção do grau de Licenciatura em Engenharia de Sistema e Informática

# Sumário

O trabalho teve como o propósito analisar o processo de desenvolvimento de um sistema de informação dentro de uma organização, no caso a Delegação do Ministério da Educação e Desporto do Concelho da Praia. Para tal, analisou-se as diferentes metodologias que estão associadas ao processo desenvolvimento de um sistema de informação, escolheu-se aplicar uma metodologia orientada a objecto.

Também no trabalho destacou-se a importância dos Sistema de Informação e Tecnologias de Informação dentro das organizações, bem como a necessidade do alinhamento dos Sistemas de Informação com as estratégias das organizações. Também analisou-se um conjunto de problemas associados ao processo de Desenvolvimento de Sistemas de informação e apresentou-se diferentes metodologias que orientam na construção de Sistemas de Informação.

**Palavras-chave:** Sistema de Informação, Tecnologia de informação, Planeamento de Estratégico de Sistema de Informação, Desenvolvimento de Sistemas de informação, Metodologias de Desenvolvimento de Sistema de Informação.

## Dedicatória

*Á minha avó Isabel Correia, por ter sido a minha inspiração nesse últimos anos. Aos meus pais pela confiança e sacrifício. Aos meus irmãos, familiares e amigos por toda a compreensão, carinho e respeito.*

# Agradecimentos

Em primeiro lugar agradeço a Deus pelo dom da vida e por todos os conhecimentos adquirido ao longo desta caminhada.

Agradeço ao meu professor e orientador, o Eng.º Stefan Manuel Monteiro, por todo o apoio e paciência durante a realização desse trabalho.

Agradeço a todos os professores do departamento de informática da Universidade Jean Piaget de Cabo verde, pelas bases de conhecimento proporcionada nesses quatro anos que serviram de apoio na realização desse trabalho.

Agradeço em especial aos meus pais Eduardo Gonçalves e Manuela Fernandes por terem acreditado em mim, e me terem encorajado a enfrentar com dignidade mais essa etapa, sobretudo por me ensinarem a respeitar os valores da vida.

Agradeço a toda meus familiares por todo o apoio, por terem colaborado para minha formação, sobretudo por terem suportado algumas das minhas ausências durante a realização desse trabalho.

Agradeço a todos os meus colegas e amigos, pelo apoio e pela amizade, também por serem compatriotas de todas as horas, o meu muito obrigado a todos.

## Conteúdo

Introdução.....	5
1 Objectivos gerais do trabalho .....	6
1.1 Objectivos específicos .....	6
1.2 Motivação do estudo.....	6
1.3 Estruturação do trabalho .....	7
Capítulo1: Desenvolvimento de Sistema de informação.....	8
1 Sistemas de informação .....	8
1.1 Conceito de Sistema de informação .....	8
1.2 Impacto dos SI/TI nas organizações.....	10
2 Planeamento Estratégico de Sistema de Informação .....	13
3 Processo de Desenvolvimento de Sistema de Informação .....	15
3.1 Problemas no processo de Desenvolvimento de Sistemas de Informação .....	16
4 Metodologias de Desenvolvimento de Sistemas de Informação .....	17
4.1 Enquadramento Histórico .....	19
4.2 Modelo em cascata .....	19
4.3 Modelo incremental .....	22
4.4 Modelo espiral .....	24
4.5 Metodologias Estruturadas .....	25
4.5.1 Structured Analysis, Design and Implementation of Information Systems .....	26
4.5.2 Yourdon System Method.....	27
4.5.3 Structured Analysis and Design Methodology .....	28
4.5.4 Méthode d’Etudes et de Réalisation Informatique pour les Systèmes d’Entreprise .....	30
4.5.5 Algumas considerações sobre as metodologias estruturadas .....	31
4.6 Metodologia da Engenharia de Informação.....	32
4.7 Metodologias orientadas a objecto .....	33
4.7.1 Unified Modelling Language .....	41
4.7.2 Rational Unified Process .....	42
4.7.3 Algumas considerações sobre as metodologias orientadas a objectos .....	46
4.7.4 Análise da metodologia Orientada Objecto versus Estruturada .....	46
4.8 Metodologias ágeis .....	48
4.8.1 Extreme Programming.....	50
4.8.2 Algumas considerações sobre as metodologias ágeis .....	53
4.9 Metodologias alternativas.....	54
4.10 Considerações sobre as metodologias Desenvolvimento de Sistemas de Informação .....	54
5 Consideração sobre o primeiro capítulo .....	56
Capítulo 2: Desenvolvimento da aplicação SIRE .....	57
1 Apresentação e objectivo da aplicação.....	57
1.1 Enquadramento da Instituição .....	57
2 Ciclo de vida do desenvolvimento da aplicação SIRE .....	59
2.1 Conceção .....	60
2.2 Elaboração .....	63



2.3	Construção .....	69
2.4	Transição: .....	73
3	Considerações sobre o segundo capítulo .....	73
	Conclusão:.....	75
	Bibliografias.....	77
	Apêndices.....	I
i.	Planeamento das fases de desenvolvimento da aplicação SIRE.....	II
ii.	Prioridade de caso de utilização .....	III

## Figuras

Figura 1: Actividades do PSI, adaptado de (Varajão, 1998) .....	14
Figura 2: Modelo em cascata.....	21
Figura 3: Modelo Incremental, fonte: (Silva e Videira, 2005) .....	23
Figura 4 Modelo Espiral, fonte (Lopes <i>et al.</i> 2005) .....	25
Figura 5: Pirâmide de EI, adaptado de (Avison e Fitzgerald, 2003) .....	32
Figura 6: Classe Funcionário.....	34
Figura 7: Objecto do mundo real.....	35
Figura 8:Herança da classe super classe funcionário .....	38
Figura 9: Polimorfismo da classe polígono .....	39
Figura 10: Composição.....	40
Figura 11: As dimensões do RUP, fonte (Silva e Videira, 2008).....	44
Figura 12: Pratica XP .....	53
Figura 13: Estrutura Orgânica de Delegação do MED do concelho da Praia .....	59
Figura 14: Diagrama Caso de utilização.....	65
Figura 15: Diagrama de sequência .....	65
Figura 16: Diagrama de classe.....	66
Figura 17: Diagrama Entidade/ Relação.....	67
Figura 18: Subsistema .....	68
Figura 19: Arquitectura do sistema .....	69
Figura 20: Página de autenticação do sistema .....	70
Figura 21: Ambiente gráfico do Director dos recursos humano .....	71
Figura 22: Submenus da categoria funcionários.....	72
Figura 23: Consulta a sobre funcionário .....	72
Figura 24: Consultas professores em Substituição .....	73

## Siglas e Abreviaturas:

**DFD** (Diagrama de Fluxo de dados)

**DSI** (Desenvolvimento de Sistemas de Informação)

**EI** (Engenharia de Informação)

**ETHICS** (Effective Technical and Human Implementation of Computer-Based Systems)

**HTTP** (Hypertext Transfer Protocol)

**ISAC** (Information Systems work and Analysis of Change)

**MERISE** (Méthode d'Etudes et de Réalisation Informatique pour les Systèmes d'Entreprise)

**PESI** (Planeamento Estratégico de Sistemas de Informação)

**RUP** (Rational Unified Process)

**SI** (Sistemas de Informação)

**SSADM** (Structured Analysis and Design Methodology)

**STRADIS** (Structured Analysis, Design and Implementation of Information Systems)

**TI** (Tecnologias de Informação)

**TIC** (Tecnologias de Informação e Comunicação)

**UML** (Unified Modelling Language)

**XP** (Extreme programming)

**YSD** (Yourdon System Method)

## Introdução

O forte impacto e a rápida evolução das TI ao longo dos tempos vêm proporcionado sucessivos desafios às organizações. Como forma de aproveitar essas evoluções muitas organizações começaram a explorar as TI como forma de melhorar a eficiência estando cada vez mais competitiva.

Como se pode ver “as organizações recorrem constantemente as redes digitais de informação para a aquisição, armazenamento, processamento, transmissão, distribuição de informações que as ajude na criação de conhecimento e riquezas, na satisfação das suas necessidades” (Almeida, 2002).

Assim sendo os SI/TI passaram a ter um papel de destaque dentro das organizações, e o sucesso dos SI depende do sucesso das actividades do DSI, considerando que os SI têm impactos em quase todos os processo organizacionais, com isso, o DSI assuma um papel importante no sucesso das organizações. Quer isso dizer que o processo de DSI de qualidade passou a ser um desafio para a comunidade de informáticos no que se refere a capacidade de resposta aos requisitos dos utilizadores/sistemas, ao requisito do tempo de execução do projecto e do custo, bem como o factor da exploração e manutenção do sistema.

Com intuito de preencher alguns dos desafios e constrangimentos que aparecem no processo de DSI surgiram ao longo dos tempos diversas metodologias para auxiliar nessa tarefa com o objectivo de garantir que os SI desenvolvidos sejam de qualidade e que tragam benefícios genéricos para as organizações e os para os seus utilizadores.

É dentro desse contexto que o presente trabalho se irá debruçar apresentando diferentes metodologias que ajudam no processo de DSI, e assim mostrar diferentes soluções quanto as metodologias e como uma pode se enquadrar melhor num dado contexto do que outra. Igualmente o trabalho compreende o estudo de como a Web pode ser um vínculo para melhorar a comunicação entre diferentes áreas de uma organização no caso a Delegação do Ministério da Educação e Desporto do concelho da Praia.

## 1 Objectivos gerais do trabalho

O presente trabalho tem como objectivo geral apresentar uma nova solução para gestão de recursos humanos à Delegação do Ministério da Educação e Desporto do concelho da Praia, utilizando a metodologia RUP no processo de desenvolvimento do sistema.

### 1.1 Objectivos específicos

Os objectivos específicos do trabalho centram-se na importância da gestão do processo de DSI dentro de uma organização, incluindo algumas características essenciais desse processo como:

- ✓ Perceber a importância dos SI/TI na melhoria da estratégia das organizações;
- ✓ Alargar o conhecimento referente a conceitos relacionados com as metodologias de DSI;
- ✓ Aprofundar o conhecimento sobre o desenvolvimento de sistemas orientados a objecto;
- ✓ Desenvolver uma aplicação *Web* para a gestão de recursos humanos da Delegação do Ministério da Educação e Desporto do concelho da Praia;

### 1.2 Motivação do estudo

A escolha do tema surgiu do interesse despertados nas disciplinas que falam do processo DSI, nomeadamente o forte impacto que os SI/TI proporcionam para a sobrevivência das organizações actuais. Juntamente como a curiosidade de aprofundar o conhecimento sobre as metodologias de DSI, para assim entender em que circunstancia utilizar uma em detrimento de outra.

Um outro factor de motivação da escolha do tema é o de conhecer mais sobre a metodologia orientada a objecto, dado que o assunto vem despertando muito interesse no seio da comunidade de informáticos.

### 1.3 Estruturação do trabalho

Este trabalho pretende analisar o processo de desenvolvimento de SI e os diferentes tipos de metodologias que lhe estão associados, dando ênfase ao processo de desenvolvimento orientado a objecto que servirá de suporte para a parte do trabalho. Encontra-se estruturada em dois capítulos e a conclusão do trabalho:

- ✓ No primeiro capítulo foi examinada o processo de desenvolvimento de SI e um conjunto de facto que lhe está associado;
- ✓ No segundo capítulo apresenta a construção da aplicação para gestão de recursos humanos para a Delegação do Ministério da Educação e Desporto do concelho da Praia recorrendo a metodologia RUP;

## Capítulo1: Desenvolvimento de Sistema de informação

---

### 1 Sistemas de informação

Antes de começar a desenvolver qualquer SI é de extrema importância que se conheça alguns conceitos chave relacionado a esse desenvolvimento. E um dos conceitos que precisa ser compreendida, é o do SI, de forma a melhor conhecer a organização e tirar maior proveito das informações que são disponíveis. Também há outros conceitos que estão envolvidos no processo de DSI que ao longo do trabalho serão também abordados.

#### 1.1 Conceito de Sistema de informação

A percepção do conceito de SI dentro de uma organização é de extrema importância para o processo de DSI, de forma a recolher informações precisas que melhor auxiliam nessa tarefa. Entretanto existe dois outros conceitos também importantes, e que estão intimamente ligados ao conceito de SI, que por sua vez precisam também ser compreendidos de forma a ajudar na compressão do conceito do SI, e no entendimento do seu papel dentro da organização, que são: dados e informação.

Dados “são elementos ou valores discretos que isoladamente não têm qualquer utilidade” (Varajão, 1998), ou seja, “são factos na sua forma bruta que representam eventos que ocorrem

nas organizações e no seu ambiente, antes de serem organizados e dispostos na forma em que as pessoas possam entender e usar” (Laudon e Laudon, 2006).

Informação são “conjuntos de dados que foram encaixados de forma que sejam significativas e úteis para o ser humano” (Laudon e Laudon, 2006), e “quando fornecidos atempadamente e de forma adequada a um determinado propósito, proporciona orientação, instrução e conhecimentos ao seu receptor, ficando este mais habilitado para decidir ou desenvolver determinada actividade” (Varajão, 1998).

Sendo assim, é possível definir SI tendo em consideração os conceitos de “dados” e “informação”. Embora não exista uma definição universalmente utilizada sobre o conceito de SI, muitos autores abordam esse conceito mostrando a forte ligação existente entre a organização e a informação.

- ✓ SI é um conjunto integrado de recursos (humanos e tecnológicos) que têm por objectivo satisfazer as necessidades de informação de uma organização e os seus respectivos processos de negócio” (Silva e Videira, 2005);
- ✓ “SI é um conjunto de meios e procedimentos cuja finalidade é assegurar a informação útil necessária às diversas funções e níveis da organização, bem como à sua envolvente externa” (Varajão, 1998);
- ✓ “SI são conjuntos de componentes inter-relacionados que colectam, processam, armazenam e distribuem informações para apoiar a tomada de decisão e o controlo de uma organização” (Laudon e Laudon, 2006);
- ✓ “SI de uma organização é o sistema que recolhe, processa, guarda e transmite informações independentemente do facto de este sistema estar informatizado ou não” (Serrano *et al.*, 2004);

O termo SI pode referir-se a um sistema que recorre ou não as TI para realização das suas tarefas, embora se reconheça que são raras as organizações actuais cujo os SI não recorrer as TI para a realização das suas tarefas. Por isso justifica-se a clarificação de do conceito de TI.



No entender de (Beynon-Davies, 2002) as TI fazem parte de SI moderno, e são meios para construir aspectos de SI onde incluem software, hardware, armazenamento e tecnologias de comunicação que tratam os dados de entrada, processam e apresentam um resultado final de forma a apoiar alguns SI.

No ver de (Serrano *et al.*, 2004) TI “é um conjunto de processos cognitivos (*software*<sup>1</sup>) e materiais (*hardware*<sup>2</sup>) necessários para a realização de uma actividade de captação, processamento, memorização ou emissão de informação”.

## 1.2 Impacto dos SI/TI nas organizações

Actualmente é visível o papel importantíssimo que os SI/TI têm vindo a proporcionar para as organizações, e os valores fundamentais que estes revelam para a sobrevivência e permanência das mesmas no mercado de trabalho, principalmente os fortes impactos sociais e estratégicos que estes lhas proporcionam. Para (Silva e Videira, 2005) “este impacto é visível não só nas grandes organizações de âmbito internacional, mas atinge também as pequenas e médias empresas”.

Entretanto, a importância do papel dos SI/TI dentro das organizações teve grandes alterações com o passar de tempo, sendo que no passado as suas perspectivas eram unicamente tecnológico, com o avançar do tempo passaram a serem vistos como factor geradores de vantagens competitivas. Conforme (Rodrigues, 2002) na década de 60, as perspectivas sobre os SI/TI eram meramente tecnológicos. Entretanto, com a revolução das aplicações durante os anos 80, houve alteração na percepção dos seus papéis, atribuindo-lhes um estatuto de favorável no posicionamento estratégico das organizações e gerador de vantagens competitivas.

Para (Almeida, 2002) “no passado, a produção de software era vista como uma actividade meramente técnica, na qual as características técnicas e funcionais eram encaradas como factores diferenciadores. Actualmente, a importância destes factores

---

<sup>1</sup> *Software*: “são instruções que controlam o funcionamento do computador, e que sem este o *hardware* não conseguia realizar as suas tarefas” (Laudon e Laudon, 1995).

<sup>2</sup> *Hardware*: “são dispositivos e meios físicos envolvidos no processamento de informação como computadores, terminais, redes físicas e outros dispositivos de armazenamento e transmissão de dados” (Varajão, 1998)

tende a diluir-se, sendo cada vez mais fácil e rápido para os concorrentes de uma empresa atingirem os níveis técnicos e funcionais que antes a diferenciavam. A qualidade dos produtos e dos serviços complementares que acompanham esses produtos, particularmente no caso de serviços a longo prazo, tornou-se no facto distintivo mais importante”.

Considerando que as organizações servem-se dos SI para obterem vantagens estratégicas e competitivas, leva a concluir que existe uma dependência mútua entre ambas as partes. De acordo com (Varajão, 1998) esse facto justifica-se devido ao “conhecimento e a compreensão de um em toda a sua abrangência, implica conhecer e entender o outro, ou seja, não é possível compreender o SI sem compreender a organização de que é parte integrante e vice-versa”.

Para (Laudon e Laudon, 1995) “o SI e as organizações têm uma influência mútua uns sobre os outros. E em contrapartida, os SI devem ser alinhados com a organização para fornecer as informações necessárias dentro da organização. E ao mesmo tempo, a organização deve estar ciente e abrir-se às influências dos SI proporcionadas pelas novas tecnologias. E assim os SI afectam a organização e as organizações necessariamente afectam a construção dos sistemas”.

Assim sendo, os SI recorrem as TI para atender as necessidades estratégicas e os objectivos das organizações. Segundo (Silva e Videira, 2005) “as TI encontram-se actualmente na origem de mudanças significativas ao nível dos modelos de negócio das empresas, e constituem um elemento fundamental para a obtenção de vantagens estratégicas e competitivas”.

Com efeito, através da utilização das TI's como factor de carácter estratégico, é possível garantir novas oportunidades de emprego, acelerar a economia, entre outros benefícios. De acordo com (Almeida, 2002) há alguns factores que demonstram o enorme potencial das TIC nas sociedades modernas como:

- ✓ O crescimento do mercado das comunicações móveis, a explosão da *Internet*, a emergência do comércio electrónico;
- ✓ A influência dos sectores das telecomunicações, dos computadores e do audiovisual.

Entretanto, entendem-se que “a interacção entre as TI e a organização é muito complexa, influenciado por um grande número de factores, incluindo a estrutura da organização, padrões de procedimentos operacionais, políticas, meio ambiente, cultura e decisão da gestão” (Laudon e Laudon, 1995).

Embora haja o reconhecimento dos valores preponderante que os SI/TI podem proporcionar para as organizações, ainda persiste algumas dúvidas dos muitos gestores perante a forma de como a organização pode obter vantagens com o aumento do investimento em novos SI/TI. No entender de (Rodrigues, 2002) há alguns dos benefícios que os SI/TI podem oferecer para as melhorar a oportunidades de negócios como:

- ✓ A redução de custos, nomeadamente através de actividades de redução do pessoal necessário;
- ✓ Aumento da produtividade, com melhoria da gestão e exploração dos recursos disponíveis;
- ✓ A melhoria no suporte, através da melhoria da informação disponibilizada aos gestores e do apoio na tomada de decisões;
- ✓ O desenvolvimento organizacional, com a utilização dos SI/TI na procura e na implementação de novos objectivos que de outra forma não poderia ser contempladas.

Reconhecendo o papel importante dos SI/TI dentro da organização, surge um novo factor da extrema importância que é a necessidade de alinhar os SI com o processo do negócio, ou seja, o sucesso das mudanças organizacionais não passam somente pelo sucesso das TI, mas principalmente pela mudança dos SI, querendo dizer que é necessário fazer um alinhamento entre os SI e a estratégia implementada pela organização, para depois conseguir obter vantagens através dos SI.

De acordo com (Lopes *et al.*, 2005) “não basta que as organizações tenham TI sofisticadas, se não souber aproveitar e integrar essas tecnologias com o alinhamento da estratégia da organização para o bem da mesma”. Com efeito, o alinhamento dos SI com a estratégia da

organização passa-se pela necessidade de se fazer um plano estratégico dos SI dentro da organização.

## 2 Planeamento Estratégico de Sistema de Informação

A utilização do PESI é de extrema importância dentro do processo de DSI, e visa identificar o conjunto de SI que tenham impacto e garantam vantagens sobre a concorrência, e que ajudem a organização a realizar a sua missões e seus objectivos.

Compreendem-se que PESI “é responsável por estabelecer as alterações ao SI em resposta á estratégia da organização (...) esta estratégia determina toda a infra-estrutura de negócio e tecnológica para suportar o desenvolvimento das actividades” (Serrano *et al.*, 2004). E de acordo com os mesmos autores (*idem*) a inexistência do PESI pode levar a consequências como:

- ✓ Perdas de vantagens competitivas perante á concorrência e deterioração da imagem perante fornecedores e amigos;
- ✓ Dificuldades da empresa em alcançar os seus objectivos, motivados por limitações do seu sistema;
- ✓ Redundâncias, imprecisões e atrasos motivados pela inexistente ou deficiente integração de sistemas;
- ✓ Conflitos entre utilizadores e responsáveis pelo desenvolvimento de SI/TI;
- ✓ Inexistência de condições para definir o nível de recursos em SI/TI requerido pela organização;

O PESI “é a parte do planeamento organizacional, incluindo pessoas, hardware e software” (Varajão, 1998), isso quer dizer, que esta tarefa “deve envolver toda a administração da organização, na análise e determinação dos objectivos que se quer para a organização, e também avaliar os possíveis modos nos quais os SI devem ser utilizados de forma que estes objectivos alcançados” (Avison e Fitzgerald, 2003).

Essa mesma ideia de envolver todos os participantes da organização no processo de PESI é também realçada por (Lopes *et al.*, 2005) “de forma a poder usar SI de uma forma estratégica, é preciso que todos os participantes da organização se envolvam em profundas considerações, de uma forma criativa e inovadora, de como construir a organização recorrendo às TI”.

Como se pode ver o PESI revela-se de muita importância para uma melhor gestão da organização, e através desta torna-se mais fácil assegurar a vantagem estratégica que tanto a organização procura. Também é necessário o PESI porque “há necessidade de justificar, racionalizar e rentabilizar os investimentos em SI/TI, quando se faz um elevado investimento e estes falham na obtenção dos benefícios esperados” (Rodrigues, 2002).

A construção do PESI passa-se pelo estudo de três actividades que são: a análise, a definição e a implementação da estratégia. Segundo (Varajão, 1998) a fase de análise tem o propósito analisar a situação actual, posteriormente na fase de definição estratégica será definida a estratégia para o futuro pretendido para a organização e os seus SI, e seguidamente na fase de implementação será definida a forma de implementação dessa estratégias, ou seja, o que se vai fazer para conseguir alcançar os objectivos traçados.

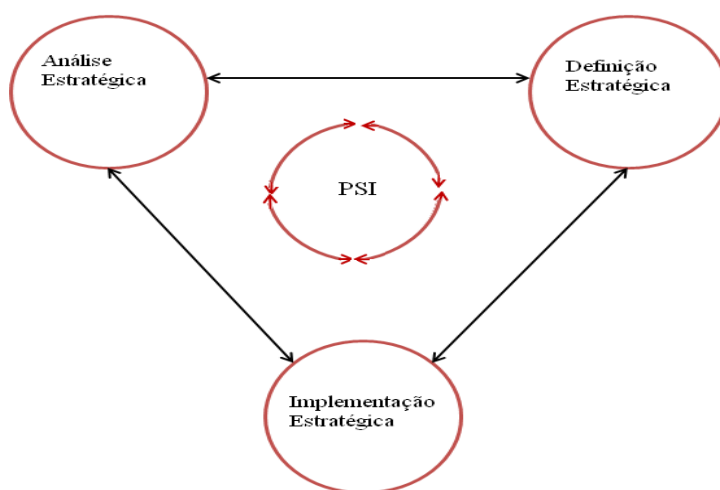


Figura 1: Actividades do PSI, adaptado de (Varajão, 1998)

Considerando o PESI de extrema importância, então as organizações procuram através do desenvolvimento deste satisfazer os seus objectivos que traduzem-se em vantagens

estratégicas. Segundo (Serrano *et al.*, 2004) as vantagens da construção de um PESI são as seguintes:

- ✓ Obter um alinhamento estratégico com o negócio (ou conjunto de actividades), procurando identificar áreas onde o investimento em SI/TI possam gerar maiores contributos e estabelecer prioridades;
- ✓ Adquirir vantagens competitivas a partir da emergência de oportunidades de negócio que seja flexível em condições económicas vantajosas;
- ✓ Desenvolver recursos e competências apropriadas para o funcionamento e para o sucesso dos SI/TI em toda a organização;

Por tudo isso é essencial que as organizações pensem correctamente nos seus SI/TI e determinem prioridades de desenvolvimento de forma a responder adequadamente à consequente inadequação dos sistemas face às necessidades actuais. E justifica-se também uma gestão correcta de SI/TI para se possa alcançar os objectivos do negócio, sabendo que há cada vez maior relevância das oportunidades e ameaças provocadas por factores externos, consequentemente através do PESI é possível assegurar que os SI desenvolvidos são utilizados de forma a obter o máximo de retorno da sua operação, ajudando a melhorar a competitividade e a produtividade da organização.

Após se ter um PESI torna-se mais fácil o desenvolvimento de qualquer SI dentro de uma organização, uma vez que com o plano todo o processo de desenvolvimento será baseado na mesma.

### 3 Processo de Desenvolvimento de Sistema de Informação

O desenvolvimento de SI visa introduzir mudanças nos SI das organizações como o principal objectivo melhorar a qualidade do seu desempenho (Avison e Fitzgerald, 2003), também é entendido como a “ciência e a arte de desenhar e fazer, com economia e elegância, SI que suportam as actividades particulares da organização” (Beynon-Davies, 2002).

Inicialmente o processo de DSI caracterizava-se por atribuir grande importância as TI no processo de DSI destinados a suportar as actividades da organização, no entanto com o avançar do tempo sentiu-se também a necessidade de envolver as pessoas nesse mesmo processo, uma vez que estas vêm proporcionando grandes impactos no processo de mudanças organizacionais.

### 3.1 Problemas no processo de Desenvolvimento de Sistemas de Informação

Nem todos os SI desenvolvidos são concluídos com êxito devido á vários problemas e incómodos que surgem ao longo do seu processo de construção. Para (Ramos, 2006) “a produção de software é, frequentemente, uma actividade não bem architectada, por vezes caótica, sem orientações de natureza e plano de gestão e controlo, o que consequentemente originam novos problemas”.

Por conseguinte, alguns dos problemas de DSI só são conhecidas após a instalação do produto onde começam a surgir novos problemas, dado que o novo sistema não satisfaz as reais necessidades do cliente, ou porque os prazos e custos do projecto ultrapassam os anteriormente estabelecidos no contrato. Conforme (Silva e Videira, 2005) há inúmeros factores que poderão estar na origem desses problemas como por exemplo:

- ✓ A falta de apoio dos gestores de topo das organizações;
- ✓ A falta de envolvimento dos utilizadores em fornecer informações necessárias para o processo de desenvolvimento do sistema;
- ✓ A incompreensão dos requisitos do sistema;
- ✓ Mudanças de requisitos do negócio o que reflecte na incapacidade de lidar com essas situações;
- ✓ Falha na gestão de projecto e deficiência no processo de desenvolvimento;
- ✓ Incapacidade de identificar riscos;

- ✓ Expectativas irrealistas, entre outros factores.

Como foi referido anteriormente ao longo do processo de DSI surgem diversas falhas ou problemas derivados de alguns factores, por isso sentiu-se a necessidade de fazer o controlo do processo de DSI para que não se tenha perdas e prejuízos desnecessários. Por isso, ao longo dos tempos formam aparecendo diferentes metodologias reconhecidas e que servem para controlar e orientar o processo de DSI.

## 4 Metodologias de Desenvolvimento de Sistemas de Informação

No processo de DSI é importante que se entenda correctamente o problema, para só depois determinar uma solução para esse problema, e sem o auxílio de uma metodologia essa tarefa torna-se mais complexa, o que por vezes conduz ao insucesso do sistema. Para (Ramos, 2006) “os problemas associados ao desenvolvimento de um software são de tal dimensão que é fundamental a definição e aplicação de princípios, regras e estratégias que conduzem a melhorias significativas em todo o desenvolvimento do projecto”.

Considerando, que há diversas falhas no processo de DSI, formam aparecendo com o passar do tempo diversas metodologias, com o intuito de ajudar na construção de sistema de qualidades, que tenham padrões de construção bem definidos, que por sua vez ajudem no entendimento do processo do negócio e que ajudem também no desenvolvimento dos SI.

Entende-se que as metodologias de DSI são “uma sequência de etapas, procedimentos, ferramentas, técnicas e notações, aplicados durante o desenvolvimento de SI” (Ramos, 2006), e que “servem como guias que apresentam a ordem das actividades que devem ser realizados dentro de cada fase de desenvolvimento e indicam as ferramentas que podem ser utilizadas para visualizar o resultado das fases” (Moraes e Chiossi, 2006). Apesar de alguns autores empregarem o termo “metodologias” e “métodos” como tendo o mesmo significado, estes apresentam algumas diferenças que precisam ser esclarecidas.

Um método é entendido como um quadro que sugerem detalhes das tarefas a serem realizadas num dado processo de desenvolvimento (Beynon-Davies, 2002), incluindo um conjunto de



técnicas, cada um escolhido adequadamente para uma determinada tarefa dentro do objectivo geral do método (Lejk e Deeks, 2002).

Entretanto, uma metodologia pode ser encarada em termos abstractos como o estudo dos métodos e em termos práticos como a lógica aplicada a uma determinada situação, ou seja, um conjunto ordenado de ideias (Lopes *et al.*, 2005).

No entanto, o uso de uma determinada metodologia de DSI requer também o uso de algumas técnicas e ferramentas adequadas para um dado contexto de desenvolvimento de sistemas, por isso é importante o entendimento dos conceitos de “técnicas” e “ferramentas” para melhor os enquadrar no contexto das metodologias de DSI.

- ✓ Técnicas são “um modo particular de fazer uma dada actividade no processo de DSI. E estas servem-se de algumas ferramentas para apoiarem nas suas aplicações” (Beynon-Davies, 2002). São também “usadas com um determinado objectivo e dão respostas a certos aspectos de DSI, podendo ser utilizados em diferentes métodos” (Lopes *et al.*, 2005). Como exemplos de técnicas têm-se: técnicas de análise de dados, técnicas de análise de processo, técnicas de análise de objectos;
- ✓ Ferramentas são “conjuntos de hardware, software, dados e tecnologias de comunicação que ajudam no processo de DSI” (Beynon-Davies, 2002). Como exemplos de ferramentas têm-se: ferramenta de interface de utilizadores que permitam construir interfaces de utilizadores sofisticadas, ferramentas para análise de sistemas (DFD, Diagrama Entidade/Relacionamento, Diagrama de transição de dados), Ferramentas de para análise de projecto (Diagrama de estrutura do sistema, Especificação de módulos);

No entanto, o conceito de ferramentas é muitas vezes confundido com o das técnicas, porém uma ferramenta é um dispositivo para ajudar na aplicação de uma técnica, e uma ou mais podem ser apropriadas a uma técnica.

#### 4.1 Enquadramento Histórico

Na década de 60 os SI eram implementados sem o apoio de uma metodologia própria, e preocupava-se em descrever soluções em termos computacionais sem dar atenção a necessidade do entendimento dos requisitos do sistema.

De acordo com (Lopes *et al.*, 2005) durante esse período houve vários problemas relacionados com o processo de DSI como por exemplo: as necessidades dos clientes não eram satisfeitas pelos dos novos SI desenvolvidos, também sistemas não eram documentados o que dificultava o processo de evolução e manutenção destes para se adaptarem as novas necessidades;

Perante essas lacunas apareceram então nos finais da década de 60 e início dos anos 70 os primeiros modelos para o desenvolvimento de sistemas e as primeiras metodologias de DSI, estas que davam ênfase aos processos designadas de metodologias estruturadas. Posteriormente, pareceram outras cujo foco eram os dados, mais tarde, seguiram as metodologias orientadas o objecto (Denis *et al.*, 2005).

Apareceram também outras que davam mais atenção aos problemas organizacionais, denominadas de metodologias orientadas a problemas, e as orientadas a pessoas (Avison e Fitzgerald, 2003).

#### 4.2 Modelo em cascata

O modelo em cascata que é também designado de “análise convencional de sistema”, “análise tradicional de sistema”, “ciclo de vida de desenvolvimento de sistema” e ainda de modelo *Waterfall*, teve uma grande influência na generalização do desenvolvimento de SI (Avison e Fitzgerald, 2003).

Este modelo apareceu no início da década de 70 e foi o primeiro paradigma de desenvolvido de SI. Existem várias versões do modelo em cascata, deferindo normalmente no número,

nome e descrição de cada fase, existências de iteração entre fases e existência de validação em cada fase (Lopes *et al.*, 2005).

Para (Kendall, 2002) o modelo em cascata está dividido em sete fases: Identificação de problemas/oportunidades/objectivos, Determinação de requisitos de informação, Análises sistema, Projecto do sistema, Desenvolvimento/Documentação do software, Teste de sistema e Implementação/evolução do sistema.

- ✓ Identificação de problemas, oportunidades e objectivos: nesta fase inicial o analista concentra-se em identificar os problemas, as oportunidades e os objectivos. Esta fase é crucial para o sucesso do resto do projecto, e é importante que o analista compreenda o negócio e para juntos com outros membros da organização definirem o problema.
- ✓ Determinação dos requisitos de informação: após a identificação do problema, segue a fase de determinação de requisitos de sistema. Nesta fase o analista recorre á amostragem e investigação de dados concretos, bem como a entrevistas, questionário, observação ou prototipagem, para a determinação dos requisitos de informações de cada utilizador envolvido.
- ✓ Análises das necessidades do sistema: nesta fase o analista utiliza algumas técnicas e ferramentas para o ajudar na implementação dos requisitos determinados na fase anterior. Uma das ferramentas muito utilizadas é o diagrama de fluxo de dados, que serve para traçar as entradas, processo, saída das funções de negócio em uma forma gráfica estruturada, e também o analista utiliza-se o dicionário de dados para listar todos os artigos de dados usados no sistema.
- ✓ Projecto do sistema: o analista utiliza as informações colectadas anteriormente para criar o projecto lógico do sistema. O mesmo irá projectar todos os procedimentos de dados de entrada de forma que quando entrarem no SI, estes estejam correctos.
- ✓ Desenvolvimento e documentação do software: nesta fase o analista trabalha junto com o programador de forma construir o sistema desejado. Visa traduzir as especificações obtidas na fase de desenho para uma linguagem de programação e também criar a documentação do sistema.

- ✓ Teste de sistema e manutenção do sistema: esta fase tem como finalidade a realização de testes antes da implementação do novo sistema, de forma a corrigir os possíveis erros antes que este seja definitivamente entregue ao cliente.
- ✓ Implementação e Avaliação do sistema: nesta fase a analista planeja a conversão na organização do sistema antigo para o novo e fazer formação de utilizadores. E a evolução consiste em fazer alterações necessárias durante a vida do sistema, quer sejam correcção de erros, quer seja alterações resultantes de novos requisitos.

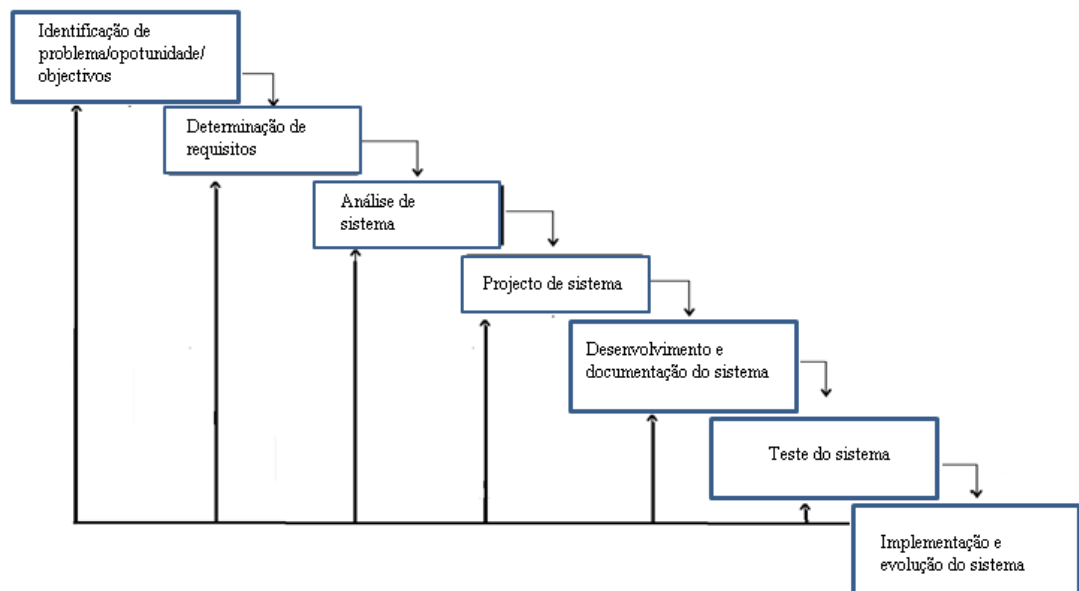


Figura 2: Modelo em cascata

O modelo em cascata tem na base da sua utilização alguns aspectos importantes. De acordo (Isaías, 2001) estes aspectos são os seguintes:

- ✓ Os sistemas são desenvolvidos do geral para o específico. Isto implica que haja acordo sobre os aspectos gerais, e se prossiga para o detalhe, evitando gastar energias em desenvolver aspectos específicos sem haver o geral.
- ✓ O conceito de sistema total é utilizado para a definição do sistema. Isto implica que o analista considere para todas as actividades da organização, os aspectos de funcionamento da mesma, quer elas sejam automáticos ou manuais.

- ✓ O método requer a comunicação entre diferentes participantes no desenvolvimento do sistema. É essencial que haja comunicação entre participantes para determinar o que se pretende obter com o sistema antes de começar a conceber o mesmo.

Embora o modelo em cascata esteja na origem de todas as metodologias de desenvolvimento de SI, no entender de (Lopes *et al.*, 2005) e (Sommerville, 2003) este modelo apresenta algumas limitações como por exemplo:

- ✓ Os acordos são feitos em estágios iniciais do processo, e isso significa que é difícil responder aos requisitos do cliente, que sempre se modificam, por isso o modelo cascata deve ser utilizado somente quando os requisitos forem bem compreendidos.
- ✓ Por vezes há uma falta de atenção prestada no contexto do negócio para o qual o sistema é desenvolvido, isso acontece porque raramente o analista questiona o porquê do desenvolvimento do sistema;
- ✓ O tempo associado com a progressão das actividades, em particular o tempo entre a especificação de requisitos e a entrega do sistema é tão grande que, por vezes, na altura de entregar o sistema os requisitos já mudaram;

#### 4.3 Modelo incremental

Dadas as limitações do modelo em cascata, apareceram então outros modelos. E o modelo incremental é um desses modelos. No entender de (Lopes *et al.*, 2005) “o modelo incremental baseia-se na ideia de que se pode construir sistemas em várias versões, cada um com um conjunto específico de sistema”.

O modelo incremental tem o seu início com a especificação não completa do processo de desenvolvimento, onde o sistema vai crescendo com o tempo, isso significa que o cliente começa por identificar um conjunto de funcionalidades que são mais prioritárias, e as que são menos prioritárias para o sistema, e a partir dessa identificação, será projectada um conjunto

de estágio de entrega, onde em cada estágio será fornecido um subconjunto das funcionalidades do sistema.

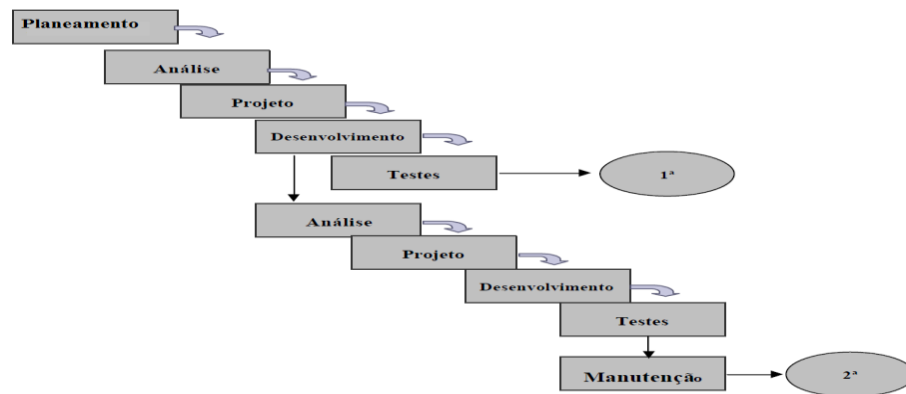


Figura 3: Modelo Incremental, fonte: (Silva e Videira, 2005)

O modelo incremental apresenta algumas vantagens, no ver de (Sommerville, 2003) as vantagens desse modelo são as seguintes:

- ✓ O cliente não precisa esperar até que todo o sistema seja entregue, para então terá proveito dele. O primeiro estágio satisfaz seus requisitos mais importantes e, assim, o software pode ser utilizado imediatamente;
- ✓ Os clientes podem utilizar os primeiros incrementos como pacotes e obter uma experiência que forneça requisitos para estágios posteriores do sistema;
- ✓ Existe um risco menor do fracasso completo do sistema. Embora possam ser encontrados problemas nos incrementos, é provável que alguns incrementos sejam entregues ao cliente com sucesso;
- ✓ Como as funções prioritárias são entregues primeiro e incrementos posteriores são integrados a elas, é inevitável que as funções mais importantes passem por maior parte dos testes. Isso significa que é menos provável que o cliente encontre falhas nas partes mais importantes do sistema;

Entretanto o modelo incremental também apresenta algumas limitações, de acordo com (Sommerville, 2003) essas limitações são:

- ✓ Os incrementos devem ser relativamente pequenos, e cada incremento deve produzir funcionalidades do sistema. Pode, portanto, ser difícil mapear os requisitos dos clientes dentro do incremento de tamanho correcto;
- ✓ A maioria dos sistemas exige um conjunto de facilidades básicas que são utilizadas por diferentes partes do sistema. Como os requisitos não são definidos em detalhes até que o incremento seja implementado, é difícil identificar facilidades comuns que todos os incrementos exijam;

#### 4.4 Modelo espiral

O modelo espiral considera o DSI como um processo contínuo, representado graficamente através de um espiral que expressa a realização de várias versões consecutivas desse sistema sistemático, no sentido de efectuar o seu constante aperfeiçoamento (Serrano *et al.*, 2004).

No entanto, o que diferencia o modelo espiral e de outros modelos de processo de DSI é a consideração dos riscos em todas as suas etapas do processo de desenvolvimento, isso possibilitará reduzir riscos antes que eles se tornem maiores. Conforme (Lopes *et al.*, 2005) este modelo define quatro actividades pelos quatros quadrantes, que são:

- ✓ Planeamento: este quadrante é responsável pela determinação de objectivos, das alternativas e restrições do projecto do sistema;
- ✓ Análise de risco: este quadrante é responsável pela avaliação de alternativas, identificação e resolução de riscos;
- ✓ Engenharia: este quadrante é responsável pelo desenvolvimento e verificação do sistema;
- ✓ Avaliação: este quadrante é responsável pela avaliação dos resultados da fase de engenharia e planeamento das fases seguintes;

No modelo espiral, cada ciclo representa uma sequência repetida de passos e começa com a identificação de objectivos e restrições. O ciclo termina com a uma revisão que abrange as pessoas da organização que requerem o SI. Esta revisão abrange todos os produtos desenvolvidos durante as fases anteriores, bem como o plano para o ciclo seguinte.

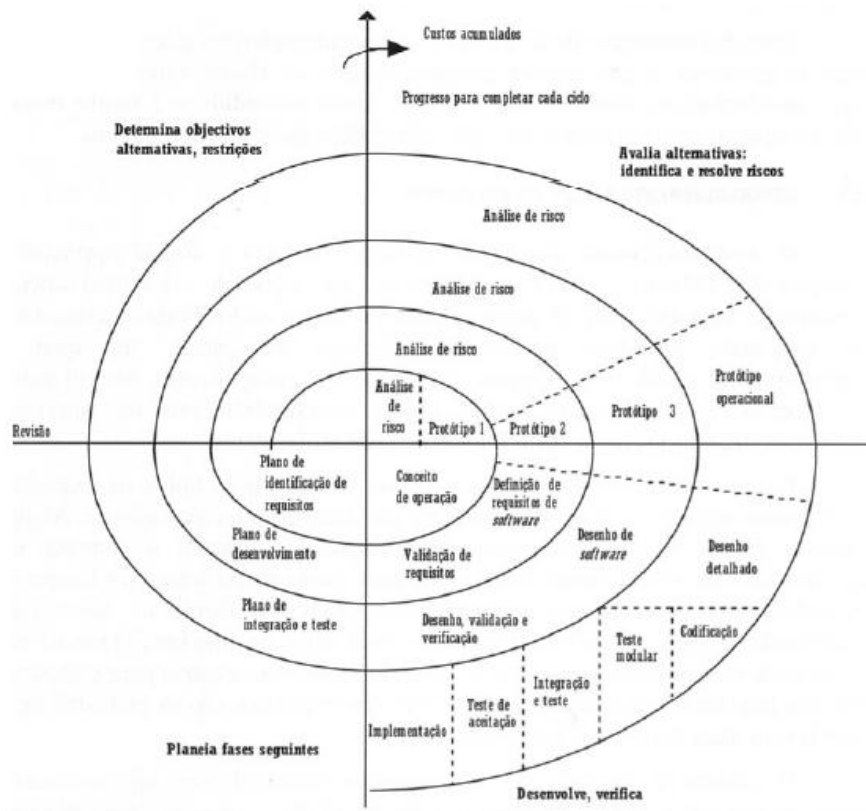


Figura 4 Modelo Espiral, fonte (Lopes *et al.* 2005)

#### 4.5 Metodologias Estruturadas

Foram desenvolvidas em meados dos anos 70 e foram as primeiras metodologias a serem utilizadas no processo de DSI. Caracterizava pela decomposição top-down do sistema, significando que dá ênfase a lógica do sistema, em vez de física, abordando assim o que os sistemas se destina a realizar (Lejk e Deeks, 2002).



A mesma foi proposta com o objectivo de otimizar as referidas fases de análise e *design* (desenho) do ciclo de desenvolvimento de sistemas, tornando mais fácil voltar ao ciclo de vida quando mudam as necessidades dos utilizadores (Isaías, 2001). Exemplo de algumas das metodologias estruturadas: STRADIS, YSM, Merise, SSADM, que passa-se a analisar.

#### 4.5.1 *Structured Analysis, Design and Implementation of Information Systems*

STRADIS foi uma desenvolvida por Gane e Sarson em finais da década de 70, baseada na filosofia de decomposição *top-down*, e é suportada essencialmente pela utilização de fluxo de dados (Silva e Videira, 2005).

Segundo (Avison e Fitzgerald, 2003) esta metodologia está dividida em quatro fases que são: estudo inicial, estudo detalhado, definição/ projecto de soluções alternativas, e projecto.

- ✓ Estudo inicial: é responsável por garantir a viabilidade do projecto, ou seja, visa garantir que os sistemas serão desenvolvidos num ambiente competitivo. Para isso, será preciso fazer uma análise de custo e benefícios de cada proposta, e então baseado nesta análise serão escolhidos as propostas que garantam o crescimento, que tenham melhores custo e melhor prestação de serviços.

Para realizar essa tarefa o analista de sistema terá que procurar descobrir essas informações juntamente com os administradores e utilizadores da organização, e também terá que analisar os documentos existentes, para só depois elaborar uma proposta que vá de acordo com a estratégia da organização.

Ao completar o estudo inicial será apresentado um produto final em relatório para a administração, que por seu lado irá decidir se vai avançar com um estudo mais detalhado ou não.

- ✓ Estudo detalhado: nesta fase o sistema será detalhado em pormenores, para isso começa-se por identificar os potenciais utilizadores do novo sistema baseando-se nas suas responsabilidades, funções departamentais, entre outros. Após identificar os utilizadores, o analista fará o levantamento de requisitos dos utilizadores recorrendo á entrevista.

Depois disso, o analista preparará o modelo lógico do sistema, onde incluirá: um DFD geral do sistema, os sistemas de interfaces, bem como um DFD detalhado para cada processo importante, e também a especificação lógica para cada processo básico a um nível apropriado de definição e as definições de dados a um nível apropriado de detalhes.

- ✓ Definição e projecto de soluções alternativas: nesta fase será definidas soluções de alternativas aos problemas do sistema existente, ou seja, os objectivos organizacionais (definidos no estudo inicial) são convertidos em objectivos do sistema. O analista baseará nesses objectivos para produzir o DFD lógico do novo sistema, e juntamente com o *designer* iram trabalhar para produzir vários desenhos alternativos para a implementação, no qual se encontre uma variável de selecção para identificar os objectivos do sistema.
- ✓ O resultado desta fase é um relatório que deve conter as seguintes informações: DFD do sistema actual, bem como as limitações do sistema actual, inclusive o custo e estimativas de benefício e o DFD lógico do sistema novo.
- ✓ Projecto físico: nesta fase a equipa de construção do sistema irá refinar a alternativa escolhida, onde envolve um conjunto de actividades paralelas em que irão detalhar os DFD que vão ser produzidos, incluindo todos os erros e excepções que não foram especificados anteriormente, bem como o dicionário de dados completo. Também será construído o projecto dos arquivos e base de dados físicos e a racionalização e a normalização

#### 4.5.2 Yourdon System Method

Esta metodologia foi proposta por Edward Yourdon, é semelhante ao STRADIS pois recorre muito à decomposição funcional, mas também atribui uma importância significativa à estrutura dos dados (Silva e Videira, 2005).

Esta metodologia não é um puro top-down nem *button-up*, mas chamado de *middle-out*, e está constituída em três fases: o estudo de viabilidade, construção do modelo essencial e construção do modelo de implementação (Avison e Fitzgerald, 2003).

- ✓ Estudo de viabilidade: esta fase consiste em fazer o estudo de viabilidade do sistema presente, do seu ambiente e os todos os problemas associados a ele. O objectivo aqui é adquirir uma compreensão geral e uma avaliação do sistema existente.
- ✓ Construção do modelo essencial: esta metodologia utiliza o termo de modelo essencial para referir o modelo lógico, e o termo de modelo de implementação quando se refere ao modelo físico (Moraes e Chiossi, 2006).

Por seu lado, o modelo essencial serve para identificar o que deve ser feito para satisfazer os requisitos dos utilizadores. O mesmo é composto por dois modelos: modelo de ambiental e modelo comportamental. Em que o modelo ambiental define a fronteira do sistema e o mundo real, e o modelo comportamental descreve o comportamento que o sistema precisa para interagir com o ambiente, produzindo as respostas requeridas por ele.

- ✓ Construção do modelo de implementação: nesta fase inicia com o processo de projecto de sistema, onde envolve um conjunto de actividades: projecto, implementação, geração de teste de aceitação, controlo de qualidade, descrição de procedimentos, conversão de base de dados e instalação.

#### 4.5.3 *Structured Analysis and Design Methodology*

Esta metodologia foi proposta por Learmonth em 1981, durante muito tempo (e ainda hoje) foi considerado a metodologia de referência, ensinada em várias universidades (Silva e Videira, 2005).

Esta se encontra dividido em cinco módulos: estudo de viabilidade, análise de requisitos, especificação de requisitos, especificação lógica do sistema, desenho físico do sistema (Lopes *et al.* 2005):

- ✓ Módulo de Estudo de Viabilidade: neste módulo será feita um estudo da área de negócio da organização, de forma a identificar se o desenvolvimento do novo sistema é viável, ou seja, se o sistema irá satisfazer os requisitos do negócio, e se tem custos aceitáveis.
- ✓ Módulo de Análise de Requisitos: neste módulo será realizada o estudo do ambiente actual onde inclui as seguintes tarefas: definir o âmbito do projecto, desenvolver um modelo de actividades de negócio, desenvolver o modelo de fluxo de dados actual, desenvolver um modelo de dados actual, investigar e reunir os resultados do estudo; Após a realização do estudo do ambiente actual, será apresentada diferentes propostas para atingir os requisitos solicitados, e como base nessas propostas será seleccionada apenas a que parecer mais favorável.
- ✓ Módulo de Especificação de Requisitos: este módulo tem por objectivo refinar e desenvolver a opção de negócio seleccionada na fase de análise de requisitos. Para tal será elaborado uma documentação com especificação detalhada que o novo sistema irá suportar.
- ✓ Módulo da Especificação lógica do Sistema: este módulo tem como objectivo definir a forma de como implementar fisicamente a especificação de requisitos da fase de especificação de requisitos, e depois será criado a especificação lógica do sistema.
- ✓ Módulo do Desenho Físico do Sistema: neste módulo será criado o desenho físico de dados bem como os programas para executar todas as funções requeridas, com base na especificação lógica definida anteriormente. Inicialmente ser criado um primeiro desenho físico, que possa ser implementado recorrendo a qualquer base de dados, sendo depois este convertido para o sistema de base de dados seleccionados.

Esta metodologia propõe a modelação de um sistema segundo três perspectivas complementares: 1-funcionalidade, 2- estrutura, 3- evolução. A primeira é representada por DFD, a segunda é obtida através do diagrama entidade associação (DEA), e a terceira pelos diagramas de ciclo de vida das entidades (Silva e Videira, 2005).

#### 4.5.4 *Méthode d'Etudes et de Réalisation Informatique pour les Systèmes d'Entreprise*

*Merise* é uma metodologia muito utilizada para o desenvolvimento de SI em França. Esta metodologia está dividida em três ciclos (Avison e Fitzgerald, 2003): ciclo de decisão, ciclo de vida, ciclo de abstracção.

- ✓ Ciclo de decisão: por vezes também chamado o ciclo de aprovação, engloba vários mecanismos de decisão, incluindo mecanismos de opções de escolha, durante o processo de DSI.

A tomada decisão é um processo que reunirá o executivo *sénior*, utilizadores e desenvolvedores de sistemas. Esta incluirá: a escolha de técnicas relativo a hardware/software, as escolhas de processamento como tempo real ou *batch*, bem com as escolhas orientados á utilizadores, relativamente à interface do utilizador. Também inclui as decisões de identificação relativo aos principais actores do sistema de informação e a organização, bem como as decisões financeiras relativo a custos e benefícios, e as decisões de administração relativo à funcionalidade dos SI.

- ✓ Ciclo de vida: esta fase mostra o progresso cronológico do SI desde a sua criação até ao fim do processo. O progresso é visto desde o inicio do seu desenvolvimento até a sua revisão final e obsolescência. As fases principais do ciclo de vida são o planeamento estratégico, estudo preliminar, estudo detalhado e a programação.

O planeamento estratégico distribui os objectivos das organizações em necessidades de informação e divide a organização em domínio para futuras análises (tal como financeiro, recursos humanos, produção). A cada uma destas divisões incluirá uma política para recursos humanos, software e produtos de hardware, e implementação de metodologia de desenvolvimento de sistema. Por seu lado o estudo preliminar descreve os sistemas de informação propostos e discute seus impactos e detalhes associados a custos e benefícios.

O estudo detalhado detalha os aspectos que serão automatizados, incluindo especificações para o projecto funcional (a especificação dos requisitos) e o projecto técnico (arquitectura técnica dos programas e arquivos), e segue a programação e

outras documentações para o desenvolvimento, bem como a implementação e manutenção do sistema.

- ✓ Ciclo de abstracção: esta fase é o ponto-chave desta metodologia. Ao contrário de alguns métodos alternativos, o tratamento separado de dados e processo é igualmente considerado.

A visão dos dados é modelada em três estágios: conceptual, lógica e físico. Em que o estágio conceptual consiste em descobrir o que o negócio faz e a essência da situação de problema, também é onde se define o que se quer fazer. Por seu lado, o estágio lógico consiste em fazer escolhas em termos da organização para o processo e modelos de base de dados para o sistema. E por fim o estágio físico consiste em introduzir assuntos relacionados com o nível operacional, sistema de administração de base de dados, e linguagens de programação.

Esta metodologia apresenta as suas vantagens (Drifa, 2003), como por exemplo:

- ✓ A sua estruturação em etapas e em pontos de controlo permite uma gestão rigorosa do projecto de concepção;
- ✓ Implementa uma linguagem comum e uma documentação bem formalizado, o que otimizar a gestão e evolução do projecto;
- ✓ Permite uma formalização do SI a níveis: conceitual, organizacional, operacional e a sua aproximação conceitual permite que não se focalize somente nas técnicas, mas acima de tudo nas métricas e nas necessidades associadas ao projecto;

#### *4.5.5 Algumas considerações sobre as metodologias estruturadas*

As metodologias estruturadas são caracterizadas por darem especial atenção aos processos. No entender de (Silva e Videira, 2005) algumas dessas metodologias apresentadas anteriormente apresentam algumas limitações, como por exemplo:

- ✓ Não conseguem lidar adequadamente com o problema da complexidade e do tamanho do crescente dos sistemas;

- ✓ Não resolvem problemas da crescente actividade manutenção do software;
- ✓ Verifica-se com frequência a má compreensão dos requisitos do utilizador, por parte dos intervenientes técnicos;
- ✓ A integração e reutilização de módulos e componentes do sistema não são fáceis;
- ✓ Os erros de concepção são descobertos tardiamente;

#### 4.6 Metodologia da Engenharia de Informação

Esta metodologia tem o seu foco centrado nos dados, ou seja, os dados desempenham o papel principal, enquanto as funções (processos) desempenham o papel secundário (Denis *et al.*, 2005).

É geralmente percebida como uma metodologia de alto nível, que pretende modelar primeiro a organização, e depois para modelar os sistemas individuais que a constituem. Esta metodologia é representada por uma pirâmide que está dividida em quatro fases (Avison e Fitzgerald, 2003), como mostra a figura5:

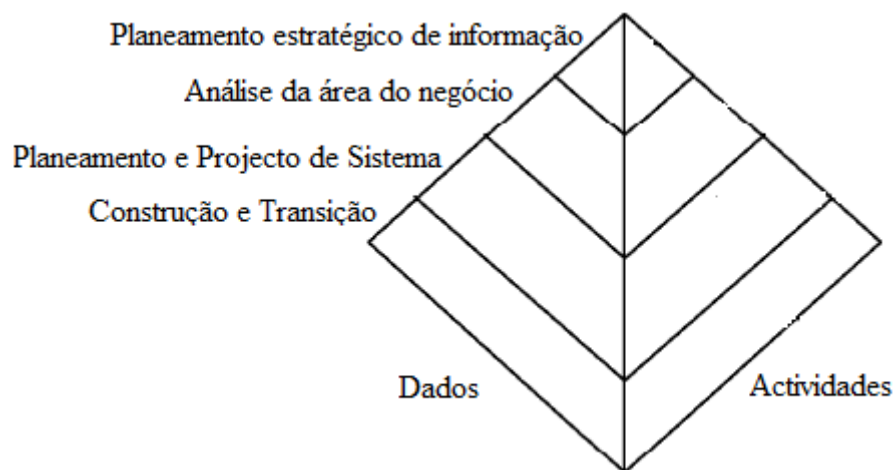


Figura 5: Pirâmide de EI, adaptado de (Avison e Fitzgerald, 2003)

- ✓ Planeamento estratégico de informação: envolve uma visão geral dos objectivos da organização, as funções de negócio e as necessidades de implementação. Esta fase

permite identificar as metas e estratégias organizacionais, delinear as funções e os seus objectivos, identificar a estrutura organizacional, estabelecer prioridades de objectivos e fornecer meios de controlo de mudanças.

- ✓ **Análise da área do negócio:** as áreas de negócio identificadas no plano de estratégia de informação passam então a ser tratadas individualmente, e busca analisar detalhadamente os dados e processos da organização.

Esta fase inclui as seguintes actividades: a criação do modelo detalhado de dados de negócio da organização, a criação de modelos de processo detalhado e vincula-o ao modelo de dados, a identificação das áreas para o projecto de sistemas. Também exige-se o envolvimento dos utilizadores do sistema e provoca uma reavaliação dos sistemas e procedimentos

- ✓ **Planeamento e Projecto de Sistema:** tem como propósito identificar como os processos seleccionados serão implementados em procedimentos e como funcionam. Para tal, são utilizadas como ferramentas os DFD, diagramas de decomposição, diagramas de estruturas de dados, *layouts* de ecrãs e *layouts* de relatórios para ajudar nessa tarefa;
- ✓ **Construção:** nesta fase o novo sistema será codificado, implementado e testado.

#### 4.7 Metodologias orientadas a objecto

O factor da produtividade em função da reutilização de componentes do sistema, bem como a construção de especificações capazes de suportar as mudanças e o desenvolvimento de sistema com maior qualidade são uns dos vários argumentos para a utilização da abordagem orientada a objecto. O conceito da orientação a objecto baseia-se na composição e interacção entre diversas unidades do mundo real chamada de objectos, com base nas suas semelhanças ao nível de comportamentos e das suas características.

O paradigma de orientação a objectos está relaciona a alguns conceitos chaves que são: objectos, atributos, comportamentos, classes, herança, encapsulamento, polimorfismo entre outros conceitos que passa-se a apresentar.



## ✓ Classes

Entendem-se que “uma classe é um grupo de partes dentro do sistema com características comuns, onde inclui informações sobre os atributos e comportamentos (operações) ” (Hamilton e Miles, 2006). Querendo dizer que as Classes são grupos de objectos com as mesmas propriedades (atributos), comportamentos (operações) e relações.

Uma Classe também pode ser entendida como “uma estrutura estática utilizada para descrever objectos mediante atributos (propriedades) e métodos (funcionalidades). A classe é um modelo ou template para criação desses objectos” (Dall’Oglio, 2009), ou ainda, “representa uma abstracção sobre um conjunto que partilham a mesma estrutura” (Nunes e O’Neill, 2004).

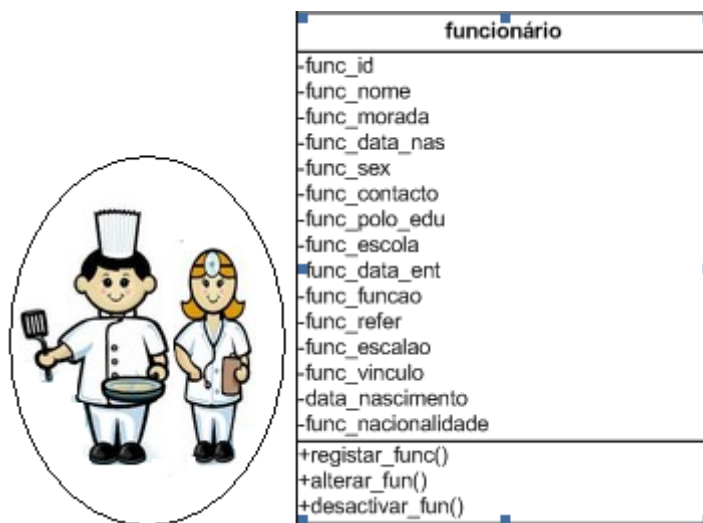


Figura 6: Classe Funcionário

## ✓ Objecto:

Não existe um conceito único para definição do que é objecto em orientação a objecto, entretanto diversos autores apresentam a sua própria definição, como se pode observar de no conceito que se seguem:

- “Um objecto é uma entidade ou conceito existente no contexto de modelação (mundo real), sendo caracterizado por um conjunto de propriedades, um comportamento e uma identidade” (Nunes e O’Neill, 2004).
- “É uma estrutura dinâmica com base em uma classe. Que após a utilização de uma classe para criar diversas estruturas iguais a ela (...) é uma instância de uma classe, porque o objecto existe durante um dado instante de tempo (da sua criação até a sua destruição) ” Dall’Oglio, 2009).
- “Um objecto pode ser um conceito, uma abstracção ou uma entidade, com fronteiras bem definidas e que tem significado no contexto de um problema e respectiva solução. Têm um estado, comportamento e identidade” (Silva e Videira, 2005).



Figura 7: Objecto do mundo real

✓ **Atributos:**

Entendem-se por atributos as características que os objectos possuem e que é representada por valor de dados. Por exemplo, o atributo Cor poderá ser igual a “vermelho” ou “Azul” (Nunes e O’Neill, 2004).

✓ **Comportamento:**

O comportamento de um objecto “é um conjunto de acções que um objecto pode realizar de forma independente” (Silva e Videira, 2005).

✓ **Encapsulamento:**

Encapsulamento nos objectos é esconder o conteúdo do objecto e apenas disponibilizar interface (operações) que fornece serviços a outros objectos, separando assim o que um objecto demonstra que faz, da forma como faz (Nunes e O'Neill, 2004).

Na percepção de (Dennis *et al.*, 2005), o encapsulamento é a combinação de processos e dados em uma única entidade. Enquanto as abordagens tradicionais de DSI tendem a serem centradas em processos ou centradas em dados, por seu lado na abordagem Orientada a objecto, os processos e dados são combinados em entidades holísticas (objectos).

Este mecanismo permite que o conteúdo do objecto possa ser alterado sem afectar os outros objectos que estão dependentes da sua interface e também aumenta a sua capacidade de reutilização. No ver de (Dall'Oglio, 2009) o encapsulamento de informação provê protecção de acesso aos membros internos de um objecto, sendo que existem certas propriedades de uma classe que devem ser tratadas exclusivamente por métodos da mesma.

O processo de encapsulamento em uma classe, é efectuado através da definição da visibilidade das propriedades e dos métodos de um objecto, que define a forma como essas propriedades podem ser acedidos. Existem três palavras reservadas que definem a forma como essas propriedades podem ser acedidas (Dall'Oglio, 2009):

- *Public* (público): membros declarados como “*public*” podem ser acedido por qualquer classe;
- *Private* (privado): membros declarados como “*private*” podem ser acedido apenas na própria classe em que formem declarados;
- *Protected* (protegido): membros declarados como “*protected*” podem ser acedido apenas na própria classe em que formem declarados e a partir das classes descendentes (herança). Como se pode ver no exemplo a seguir.

```
public class funcionario{  
    public $func_nome;  
    public $func_morada;  
    public $data_nascimento;  
    public $func_sexo;  
    public $func_nacionalidade;  
    protected $func_telefone;  
    protected $func_email;  
}
```

#### ✓ **Herança:**

O mecanismo de Herança é uma das grandes vantagens da orientação a objecto. E “consiste em permitir que estruturas comuns sejam compartilhadas entre classes, derivados e similares, sem redundância” (Ramos, 2006).

Na perspectiva (Dall’Oglio, 2009) a herança em orientação a objectos é compartilhar atributos e comportamentos entre as classes de uma mesma hierarquia.

As vantagens da Herança reside no facto da redução de linhas de códigos, ou seja, estes podem ser partilhados através de classes com comportamento comuns, o que reduz também o número de casos que devem ser entendidos e analisados, o que permitir a criação mais rápida de novas classes, também permite menos redundância. Conforme (Dall’Oglio, 2009) “a possibilidade de reutilizar de partes do código já definidos é o que dá maior agilidade no dia-a-dia, além de eliminar a necessidade de eventuais duplicações ou reescritas de código”.

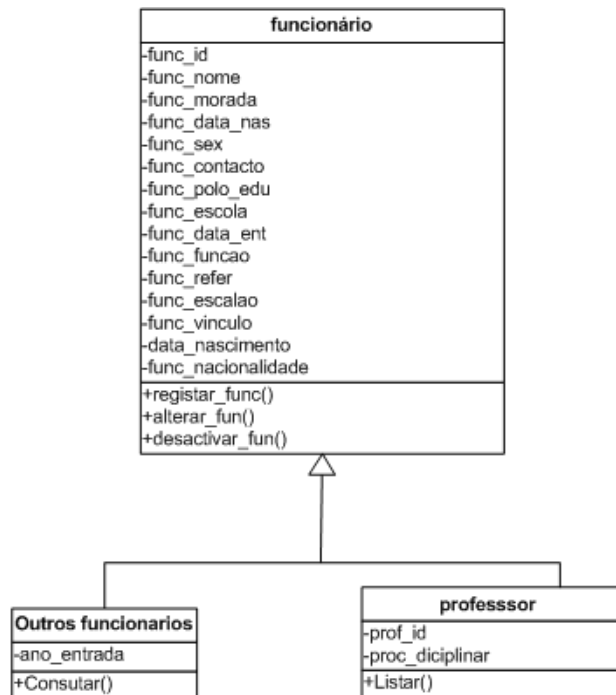


Figura 8:Herança da classe super classe funcionário

#### ✓ Polimorfismo:

Polimorfismo “trata-se da possibilidade de um objecto se manifestar de diferentes formas. Refere-se a capacidade de uma mesma operação realizar funções diferentes de acordo com o objecto que a recebe e com os parâmetros que são passados” (Ramos, 2006).

No entender de (Dall’Oglio, 2009) o polimorfismo “é o princípio que permite que classes derivadas de uma mesma super-classe tenham método iguais (com a mesma nomenclatura e parâmetros), mas comportamentos diferentes, redefinidos em cada uma das classes-filha”.

A vantagem de polimorfismo deve-se ao facto de proporcionar maior facilidade na programação de eventos, também facilita a substituição ou mudanças de objectos dentro do sistema (Dennis *et al.*, 2005).

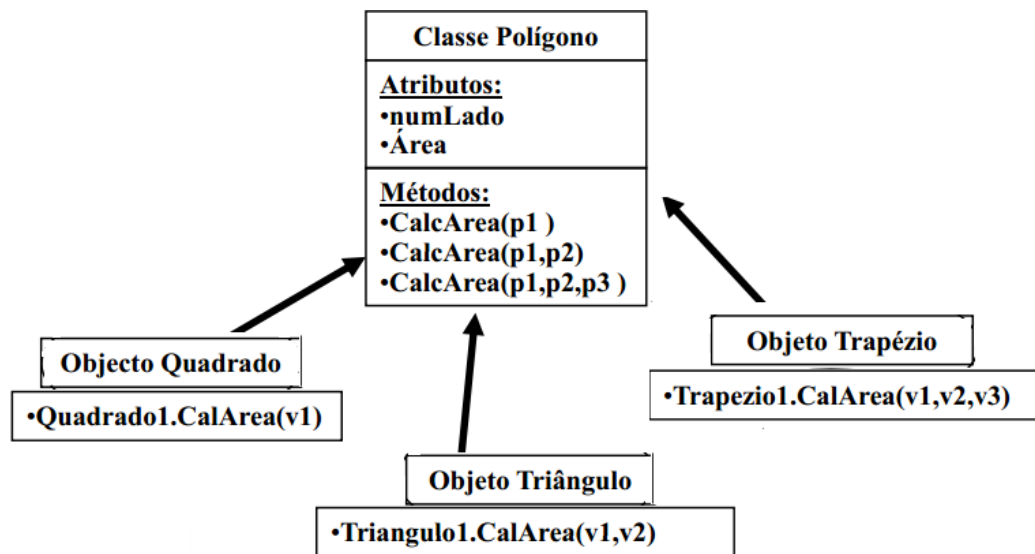


Figura 9: Polimorfismo da classe polígono<sup>3</sup>

#### ✓ **Abstracção:**

A abstracção consiste em concentrar no que um objecto é, e o que ele faz, antes mesmo de decidir como ele será implementado (Ramos, 2006).

Compreende-se que na construção de sistemas orientado a objecto, deve-se separar o sistema em partes, concentrando-se nas peças mais importantes e ignorando os detalhes (em primeiro momento), para poder construir peças bem definidas que possam ser reaproveitadas mais tarde (Dall'Oglio, 2009).

Classes abstractas “são classes que nunca serão instanciadas na forma de objectos, somente suas filhas serão” (Dall'Oglio, 2009).

#### ✓ **Associação, agregação e composição:**

Associação no diagrama de classe “representa relações entre objectos. E a agregação no diagrama de classes pretende demonstrar o facto de que um todo é composto por partes” (Nunes e O'Neill, 2004).

<sup>3</sup> Figura reiterada dos acetatos das aulas do professor (Da Luz, 2011)

Composição é “uma agregação com um significado mais forte existindo uma dependência directa entre duas classes (se a parte deixar de existir, o todo deixa de existir e vice-versa)” (Nunes e O’Neill, 2004).



Figura 10: Composição

Após abordar alguns conceitos chave relacionados da orientação a objecto, vai-se também analisar o ciclo de vida de desenvolvimento de sistema orientação a objecto. Segundo (Hoffer *et al.*, 2002) esse ciclo de desenvolvimento consiste em desenvolver uma representação de objectos em três fases: análise, desenho e implementação.

- ✓ **Análise:** nesta fase é desenvolvida um modelo da aplicação de real mundo e é exibido as suas propriedades importantes. Resume conceitos do domínio de aplicação e descreve o que o sistema tem que fazer, ao invés de como será feito. O modelo especifica o comportamento funcional do sistema, independente de preocupações relativo ao ambiente no qual este será implementado. O modelo de análise deve capturar essas exigências com precisão, pois é mais fácil e mais barato de fazer mudanças ou fixar falhas durante a fase de análise do que durante as fases posteriores.
- ✓ **Desenho:** esta fase define como o modelo de análise será percebido no ambiente de implementação. Durante esta fase é construído com mais detalhes o modelo de desenho que será utilizado na implementação, como a reestruturação de classes, dados da estrutura interna e algoritmos para implementar em cada classe, implementação das associações, implementação de controlo, adicionados a um modelo físico, para cada modelo de análise conforme a estratégia estabelecida durante o desenho do sistema.

- ✓ Implementação: nesta fase será implementada o desenho do sistema usando uma linguagem de programação e um sistema de administração de base de dados, ou seja, visa traduzir a fase de desenho em código de programas, e esse processo é relativamente directo, dado que o modelo de desenho já incorpora a linguagem de programação

Após analisar o ciclo de vida de um sistema orientado a objecto vai se analisar também algumas metodologias associado a esse paradigma de desenvolvimento de sistemas. De acordo com (Silva e Videira, 2005) as metodologias orientadas a objecto tiveram grandes repercussões ao longo das décadas de 80 e 90, onde surgiram nessas décadas inúmeras propostas metodológicas que utilizavam conceitos relacionados com a orientação a objecto.

Exemplos destas metodologias: OOSE (*Object Oriented Software Engineering*), OOAD (*Object Oriented Analysis and Design*), OMT (*Object Modelling Technique*), RUP. Seguidamente vai-se analisar alguns conceitos relacionados com estas metodologias e analisar a metodologia RUP.

#### 4.7.1 *Unified Modelling Language*

A UML não é uma metodologia de DSI, mas sim uma linguagem de modelação de sistemas, que utiliza notação padrão para especificar, construir e documentar SI orientadas a objectos.

Devido a proliferação de métodos e notações para modelação orientada a objecto sentiu-se a necessidade de ter uma nomenclatura comum para a modelação desse SI. Segundo (Nunes e O'Neill, 2004) devido a essas inúmeras propostas e cientes da necessidade de ter um padrão universalmente aceite, então alguns autores começaram a trabalhar juntos e apresentaram uma proposta unificadora dos seus trabalhos individuais, que originou a UML.

O modelo em UML é constituído por um conjunto de diagramas que representam aspectos complementares de um SI (Nunes e O'Neill, 2004). Segundo esses autores os principais diagramas utilizados são os seguintes:



- ✓ Diagrama de casos de utilização (*Use Case*): permite a captação dos requisitos funcionais do sistema, ou seja, serve para identificar as fronteiras do sistema e descreve os serviços (*use cases*) que devem ser disponíveis a cada um dos diversos utilizadores (actores);
- ✓ Diagrama de classes: permite descrever a estrutura de informação (classes e relação) que são utilizadas no sistema;
- ✓ Diagramas de objectos: que pode ser utilizado para ilustrar diagramas de classes com exemplos concretos, ou seja, mostra os objectos que formam instanciados pela classe;
- ✓ Diagramas de sequências e colaboração: que servem para ilustrar como os objectos do sistema interagem para fornecer a funcionalidade do *use case*;
- ✓ Diagrama de estado: é utilizado para modelar o comportamento dos objectos, isto é, descreve alterações nos valores dos atributos dos objectos em resultado da ocorrência de certos eventos;
- ✓ Diagrama de actividade: pode ser utilizado para descrever cada um dos *use cases*, realçando o encadeamento por cada um dos objectos do sistema;
- ✓ Diagrama de componentes: utilizados para descrever a arquitectura da aplicação informática em termos de componentes de software, ou seja, mostra como as classes deverão se encontrar organizadas através da noção de componentes de trabalho;
- ✓ Diagrama de instalação: permite descrever a arquitectura de equipamento informático utilizado e atribuição dos componentes da aplicação aos diversos equipamentos;

#### 4.7.2 *Rational Unified Process*

O RUP foi desenvolvido na década de 90, com o objectivo de assegurar a produção de sistemas com qualidades e de poder ser utilizados em um grande número de projectos e organizações (Lopes *et al.*, 2005).

No entender de (Kroll e Kruchten, 2003) o RUP cria processo bem definidos e bem estruturado, pois define claramente quem é responsável por o quê, como as coisas serão terminadas, e quando fazer.

Esta metodologia baseia-se em boas práticas de desenvolvimento de software. Conforme (Kruchten, 2004) e (Silva e Videira, 2008), estas características são as seguintes:

- ✓ Baseado em caso de uso: utiliza-se diagramas de caso de utilização (*Use Case*) para capturar requisitos funcionais do sistema, e esta conduz ao próprio desenho, implementação e teste do sistema. Onde durante o período de teste vai se verificar se o novo sistema realiza o que está escrito diagrama de caso de utilização, também os casos de utilização são utilizados para planear e acompanhar as interações;
- ✓ Centrada numa arquitectura: visão geral em termos dos seus subsistemas e como estes se relacionam;
- ✓ Interactiva e incremental: realiza o processo de desenvolvimento de sistema é baseado em iterações, e de uma forma iterativa, em que o conhecimento do sistema vai-se crescendo com as iterações, onde o sistema será refinado e a solução vai aparecendo á cada incremento, e assim fica mais rápido detectar riscos e minimiza-los;
- ✓ Gestão integrada de requisitos: o processo RUP tenta fazer uma gestão integrada dos requisitos desde a sua identificação até à sua implementação, facilitando a demonstração do facto do sistema satisfazer as necessidades do utilizador;
- ✓ Modelação visual: que facilita a comunicação entre elementos da equipa e promove diferentes níveis de abstracção, também permite entender melhor não só a concepção e complexidade do sistema, mas também facilita a identificação e resolução do problema;
- ✓ Verificação de qualidade do software: é uma tarefa que deve ser feita não só no final do projecto, ou relegando responsabilidade do mesmo para equipa independente, mas fazendo parte integrante do projecto, e construindo-se como preocupação permanente de toda a equipa;

- ✓ Controlo das alterações ao software: definindo um processo que formaliza a gestão de um dos factores que coloca muitas vezes em risco o sucesso de um projecto, mas no qual se investe frequentemente pouco tempo de esforço;

O processo do RUP é descrito em duas estrutura (ou dimensões) a componente estática e a componente dinâmica conforme ilustrada na figura 11. Segundo (Kroll e Kruchten, 2003) estas duas dimensões podem ser descritas como o seguinte:

Componente dinâmica (visão dinâmica) que corresponde a dimensão horizontal e representa a estrutura dinâmica ou a dimensão do tempo do processo expressada em ciclo, fases, iterações e marcos. Desdobra sobre o ciclo de vida do projecto;

A componente estática (visão estática) correspondente a dimensão vertical e que representa a estrutura estática do processo, ou seja, as actividades e tarefas realizadas, disciplinas, artefactos e papéis de todos os intervenientes envolvidos no projecto;

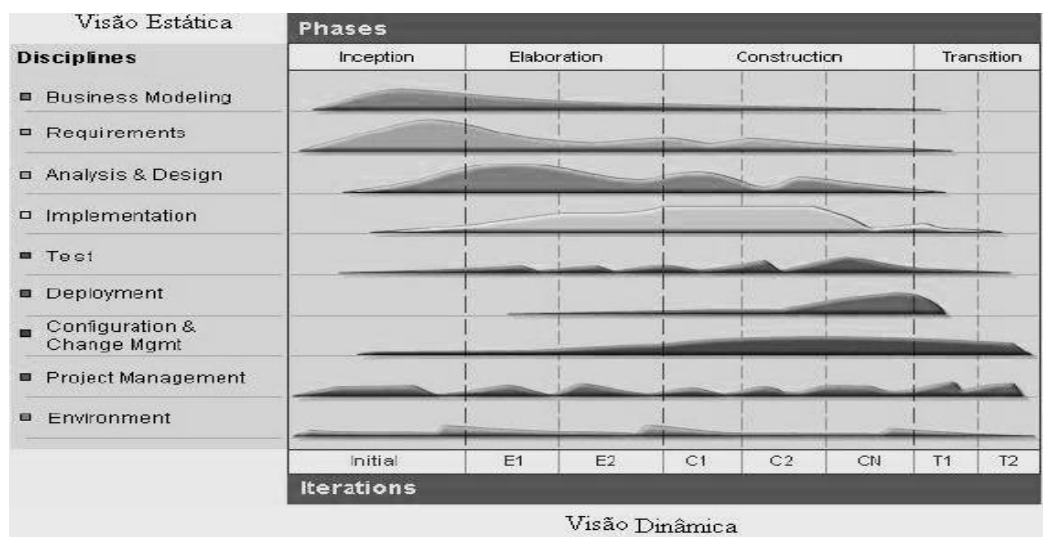


Figura 11: As dimensões do RUP, fonte (Silva e Videira, 2008)

Por seu lado, os componentes estáticos (visão estática) correspondente a dimensão horizontal, descrevem as tarefas e as actividades realizadas, bem como os produtos de trabalho e os

respectivos intervenientes. Encontram-se centradas no conceito de disciplinas que são as seguintes:

- ✓ Modelação do negócio (*Business Modeling*): inclui as actividades necessárias para compreender os processos de negócio da organização;
- ✓ Requisitos (*Requirements*): agrupa o conjunto de actividades relacionadas com a identificação e modelação dos requisitos do sistema;
- ✓ Análise e Desenho (*Analysis and Design*): actividades de definição da solução e de como o sistema será construído na fase de implementação.
- ✓ Implementação (*Implementation*): actividades de produção de código e testes unitários.
- ✓ Teste (*Test*): verificação de actividades de todo o sistema.
- ✓ Instalação (*Deployment*): actividades de disponibilização do sistema para os seus utilizadores finais

O ciclo de desenvolvimento do projecto RUP correspondente a componente dinâmica é identificado através de quatro fases (Kroll e Kruchten, 2003). Cada fase com respectivos marcos de finalização também definidos de *milestones*. Os *milestones* são indicadores de progresso de projecto, e são usadas como base para decisões para continuar, abortar, ou mudar o rumo do projecto (Silva e Videira, 2008). As fases do RUP são as seguintes:

- ✓ Concepção (*Inception*): tem por objectivo entender a dimensão do projecto, obter uma visão global do sistema, eliminar riscos mais importantes e efectuar a definição do âmbito do projecto.
- ✓ Elaboração (*Elaboration*): tem como objectivo especificar as funcionalidades do sistema, ou seja, esta fase pretende analisar o domínio do problema, desenvolver o plano do projecto.
- ✓ Construção (*Construction*): pretende implementar e testar o software.

- ✓ Transição (*Transition*): pretende distribuir o produto ao cliente final, bem como efectuar todas as actividades necessárias para garanti o respectivo sucesso.

#### 4.7.3 *Algumas considerações sobre as metodologias orientadas a objectos*

Embora se reconheça as grandes vantagens que a orientação a objecto traz no processo de DSI, como o factor da reutilização e a manutenção, esta metodologia apresenta algumas limitações. Na perspectiva de (Silva e Videira, 2005) essas limitações são os seguintes:

- ✓ Reconhece-se que nem sempre é fácil encontrar os objectos e classes apropriados no domínio do problema, já que a maioria dos informáticos continuam a pensar em termos funcionais;
- ✓ Só recentemente começaram a surgir no mercado ferramentas de apoio ao processo de desenvolvimento segundo o paradigma orientado a objecto;

#### 4.7.4 *Análise da metodologia Orientada Objecto versus Estruturada*

Analisando o processo estruturado e a orientação a objecto, há um alguns de factores que permite reconhecer vantagens da orientação a objecto face a estruturada. De acordo com (Silva e Videira, 2005) estas vantagens podem ser as seguintes:

- ✓ No desenvolvimento estruturado, o sistema consiste num conjunto de dados que são usadas por inúmeras funções, de forma mais ou mesmo interdependentes, enquanto no desenvolvimento orientação a objecto, os dados e função são agregados conjuntamente numa entidade lógica (objecto) que providencia uma interface bem definida para outros objectos comunicarem entre si;
- ✓ Na abordagem orientada a objecto, as entidades do mundo real são representadas directamente por objectos, que apresentam dados e comportamento, bem como a entidade própria. Por seu lado na abordagem estruturada, as entidades são representadas através de módulos com funções específicas que acedem a várias estruturas de dados de forma interdependentes.

- ✓ O desenvolvimento orientação a objecto providencia blocos de construção de alto nível que reduzem custos de desenvolvimento, pela promoção da reutilização e encapsulamento de software.

Com efeito, é possível afirmar que a metodologia estruturada traz vantagens quando aplicada em aplicações pequenas, onde se pressupõe que não haverá introduções de mudanças porque só desta forma é mais fácil de ler e compreender o código, porém a partir do momento em que for necessário introduzir novos módulos ou adicionar recursos, a metodologia estrutura pode ser um problema, pois o sistema fica mais complexo e será necessário repetir códigos já escritos, e assim surge uma das grandes vantagens das metodologias orientadas a objecto face a metodologias estruturadas, que é o processo de manutenção do sistema.

Todavia, a metodologia orientada a objecto trabalha com problema de maior grau de complexidade, pois permite construir especificações capazes de resistir a mudanças e reutilizar resultados anteriormente realizados. No ver de (Serrão e Marques, 2007) a abordagem orientada a objecto apresenta vantagens significativas como:

- ✓ A reutilização: a orientação a objecto permite que os objectos sejam combinados de diversas formas, fazendo com que possam ser reaproveitados, e assim ajuda a poupar o tempo de desenvolvimento;
- ✓ Expansibilidade: ao invés de escrever um objecto totalmente novo sempre que necessário, pode ser mais preferível estender algum já existente, ou seja, é possível derivar um objecto de outro objecto, estendendo as suas funcionalidades sem ter que reescrever o objecto com funcionalidade adicionais;
- ✓ Manutenção: devido à forma natural dos objectos e à forma como estes formam concebidos, torna-se muito mais simplificado a sua manutenção;

Apesar de existirem muitos factores que apontam grandes vantagens das metodologias orientadas a objecto em relação as estruturadas, há casos em que a metodologias estruturas se enquadra melhor. Segundo (Silva e Videira, 2005) se o ambiente de desenvolvimento for

estruturado, ou se a linguagem de programação for Cobol ou C, então fará mais sentido utilizar a metodologia estruturada de que orientada a objecto.

#### 4.8 Metodologias ágeis

Na procura de desenvolver SI com mais qualidade, surgiu então uma nova metodologia designada de metodologias ágeis, que apresenta um conjunto de valores e boas práticas que considera fundamental no processo de DSI, buscando eliminar o tempo gasto na documentação do sistema, para criar sistema mais iterativo, mais simples e com melhor qualidade.

A mesma teve a sua origem no ano 2001, em Utah nos EUA, onde reuniram-se 17 pessoas de renome ligados a área de SI, com o intuito de partilharem as suas visões sobre o processo de DSI. E do encontro resultaram a constituição relativamente informal de um grupo, com definição de pontos comuns através de manifesto de desenvolvimento ágil de software, baseado num conjunto de valores e princípios. Estes defendem a não realização de actividades de análise e desenho, e a produção de menos documentação, e concentram-se nas necessidades das pessoas e não na definição do processo<sup>4</sup>. Os principais valores das metodologias ágeis<sup>5</sup>:

- ✓ Indivíduos e interacção ao invés de processos e ferramentas: é mais importante a pessoa e os aspectos relacionados com a motivação, ambiente de trabalho, ritmo de trabalho, qualidade de vida, assim como a forma como as pessoas trabalham.
- ✓ Software que funciona ao invés de documentação perceptível: é mais importante a construção de um sistema que funciona do que produzir toda a documentação sobre o sistema cujo desenvolvimento nunca chega ao fim ou que falha na concretização dos seus objectivos.
- ✓ Colaboração com cliente ao invés de negociação de contrato: é mais importante estabelecer uma relação de colaboração e confiança com o cliente do que gastar tempo em discussões sobre as cláusulas do contrato.

---

<sup>4</sup> Disponível em <http://www.agilemanifesto.org/history.html> Consultado em Setembro 2011

- ✓ Responder as mudanças ao invés de seguir com plano: é mais importante responder as alterações expectáveis dos requisitos do que seguir um plano de projecto previamente definido, que constantemente é alterado;

Princípios das metodologias ágeis<sup>6</sup>:

- ✓ A maior prioridade é satisfazer o cliente através da entrega antecipada e contínua de software funcional;
- ✓ Os requisitos de constante mudanças são sempre bem-vindos, mesmo no final do desenvolvimento. Ágeis baseiam-se nos processos de mudanças para ganhar a vantagem competitiva do cliente;
- ✓ Entregar software funcional com frequência, a partir de duas de semanas ou de dois meses, com preferência para a escala de tempo mais curto;
- ✓ Homens de negócio e desenvolvedores de sistemas têm que trabalhar juntos diariamente durante a construção do projecto;
- ✓ Construir projectos com equipas motivados. Proporcionar-lhes o ambiente e suporte que precisam, e também confiar neles para fazer o trabalho;
- ✓ O método mais eficiente e eficaz de transmitir informações dentro do processo de desenvolvimento é a conversa cara-a-cara;
- ✓ Trabalhar o software é a principal medida de progresso;
- ✓ Os processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvidos e utilizadores devem ser capazes de manter um ritmo constante indefinidamente;
- ✓ Atenção contínua à excelência técnica e bom *design* aumenta a agilidade;
- ✓ Simplicidade: a arte de maximizar a quantidade de trabalho não feito, é essencial;

---

<sup>5</sup> Disponível em <http://www.agilemanifesto.org/> Consultado em Setembro 2011

<sup>6</sup> Disponível em <http://www.agilemanifesto.org/principles.html> Consultado em Setembro 2011



- ✓ As melhores arquitecturas, requisitos e projectos emergem equipas auto-organizadas;
- ✓ Em intervalos regulares, a equipa reflecte sobre como se pode tornar mais eficaz, depois sintoniza e ajusta seu comportamento em conformidade;

O desenvolvimento ágil utiliza uma abordagem colaborativa, incremental e interactiva para desenvolvimento de software que pode produzir software de alta qualidade, a um custo eficaz e a tempo. Ao contrário de desenvolvimento de software tradicional, o desenvolvimento ágil enfatiza o teste de flexibilidade, integração contínua e a entrega rápida de software<sup>7</sup>. Exemplo de algumas metodologias ágeis: XP, Scrum, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development.

#### 4.8.1 *Extreme Programming*

Um dos grandes objectivos do XP é reduzir o custo de alteração de requisitos, e para tal propõe um conjunto de valores, princípios e boas práticas para o desenvolvimento de sistemas (Dennis *et al.*, 2005). Esta metodologia está assente em quatros valores<sup>8</sup> importantes que são:

- ✓ Comunicação: é importante a comunicação entre todos os interessados no projecto de desenvolvimento de software, para criar a melhor solução para a resolução do problema.
- ✓ Simplicidade: o desenho e o código devem ser orientados a satisfação dos requisitos actuais e não para os que possam surgir numa próxima interacção. Fazer o que for necessário e o solicitado, não mais. Isto irá maximizar o valor do investimento feito á a data. E assim desenvolver algo que o cliente goste e mantê-lo a longo prazo com custos razoáveis.
- ✓ Feedback: Tomar todos os compromisso sérios, fornecendo software cedo e em constante iteração, prestando atenção as mudanças solicitadas pelo cliente, ou seja, interagir com o cliente para tornar o produto adaptável a necessidade do mesmo.

---

<sup>7</sup> Disponível em <http://www-01.ibm.com/software/rational/agile/> Consultado em Setembro 2011

<sup>8</sup> Disponível em <http://www.extremeprogramming.org/values.html> Consultado em Setembro 2011

- ✓ Coragem: a coragem permite que os utilizadores se sintam confortáveis a refazer o código sempre que necessário, isto é, sempre que se verifique as alterações futuras possam ser implementadas mais facilmente. Dizer a verdade sobre os progressos e estimativas. Trabalhar para adaptar as mudanças quando elas aparecem e tirar lições aprendidas sobre a implementação de XP até o momento.
- ✓ Respeito: o respeito e sentido de responsabilidade manifestam-se de várias formas. No respeito pelo trabalho de equipa, pelos seus colegas e pelos compromissos assumidos. Sentimento de respeito por todos da equipa do projecto, desenvolvedores respeitem a experiencia do cliente e vice-versa. Respeito para a autoridade e responsabilidade do projecto.

Entretanto o XP apresenta um conjunto de princípios que são:

- ✓ Equipa Coesa (*Whole Team*): A equipa de desenvolvimento é formada pelo Cliente e pela equipa de desenvolvimento;
- ✓ Jogo de planeamento (*The planning Game*): Define-se rapidamente o âmbito da próxima release de desenvolvimento através de um “jogo” disputado pelos programadores, em que os primeiros definem prioridades de negócio e os últimos estimam o esforço e a concretização;
- ✓ Pequenas *releases* (*Small Releases*): Disponibilizar o mais rápido possível uma versão simplificada em produção, sendo posteriormente e continuamente disponibilizadas novas versões do sistema em ciclo de tempo curto;
- ✓ Teste de cliente (*Customer Tests*): cliente constrói testes, e um conjunto de analistas e testadores trabalham para aceitação dos determinados requisitos do sistema;
- ✓ Desenho simples (*Simple Design*): o sistema deve ser desenhado e implementado, em cada momento, da forma mais simples possível, devendo ser eliminado qualquer complexidade não necessária. Equipas de XP constroem *software* para projectos simples, mas adequada;

- ✓ Refatorização (*Refactoring*): o processo de refatorização concentra-se na remoção de duplicação, e no aumento da “coesão” do código, enquanto diminui o “acoplamento”. Os programadores reestruturam continuamente o sistema sem alterar o seu comportamento, de forma a remover código duplicado, melhorar a comunicação, simplificar, flexibilizar ou generalizar o código;
- ✓ Programação em par (*Pair Programming*): o código é construído por pares de dois programadores, que trabalham lado-a-lado no mesmo computador, um escreve o código enquanto o outro verifica a correcção. Esta prática garante que todo o código produzido é revisto por pelo menos um outro programador, e resulta em melhor design, melhor teste e melhor código.
- ✓ Desenvolvimento conduzido por testes (*Test-Driven Development*): XP é baseado em feedback, e no desenvolvimento de sistema, um bom feedback exige um bom teste. Os programadores escrevem continuamente testes unitários, antes de escreverem o código propriamente dito, cuja execução garante automaticamente a correcção do sistema.
- ✓ Integração continua (*Continuous Integration*): A equipa de XP mantém o sistema integrado em todos os momentos. O sistema deve ser integrado e compilado várias vezes ao dia, sempre que uma tarefa é concluída e integrada no sistema.
- ✓ Propriedade colectiva do código (*Collective Code Ownership*): o código pertence a toda a equipa, pelo que qualquer programador pode alterar parte do código em qualquer altura oportuno ou necessário;
- ✓ Padrão de codificação (*Coding Standard*): Equipas de XP seguem um padrão comum de codificação, de modo que todo o código no sistema esteja como se fosse escrito por um;
- ✓ Metáfora (*Metaphor*): a metáfora é uma descrição simples de como o programa funciona. Define-se histórias simples e partilhada, que guia todo o desenvolvimento e ajuda a explicar a visão geral do sistema;

- ✓ Ritmo de trabalho Sustentado (*Sustainable pace*): Trabalhar com qualidade, buscando ter sempre um ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem hora extras. As horas extras são bem-vindas quando trazem muito valor para o desenvolvimento;

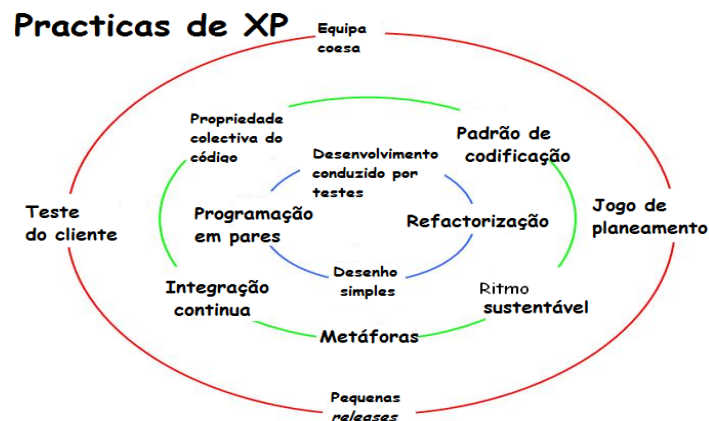


Figura 12: Pratica XP<sup>9</sup>

#### 4.8.2 Algumas considerações sobre as metodologias ágeis

A metodologia ágil apesar de proporcionarem maior agilidade no processo de desenvolvimento de sistema, a sua aplicação deve ser contextualizada, pois a mesma apresenta algumas limitações. Segundo (Silva e Videira, 2008) essas limitações são as seguintes:

- ✓ Os projectos ágeis apresentam algumas dificuldades quando utilizados em projectos de média e grande dimensão, quer em termos de tempo, pessoas envolvidas ou mesmo o âmbito do projecto;
- ✓ Não são recomendados para projectos críticos ou complexos, ou que tenham requisitos relativamente bem definidas e estáveis;

<sup>9</sup> Disponível em <http://www.xprogramming.com/what-is-extreme-programming/> Consultado em Setembro 2011

#### 4.9 Metodologias alternativas

Existem outras metodologias de DSI que dão mais importância a aspectos sociais, como é o caso de ETHICS e há outras que se preocupam com os problemas das organizações é o caso da metodologia de ISAC.

O ETHICS combina o aspecto sócio com os aspectos técnicos, ou seja, têm uma visão tecno-social de que as tecnologias devem atender os factores organizacionais e sociais (Kendall, 2002).

ETHICS é uma metodologia orientada á pessoas, baseado na participação das pessoas no processo de DSI. Assim sendo os utilizadores são envolvidos nas decisões relativamente ao processo de trabalho e como o uso das tecnologias poderia melhorar a satisfação dos seus trabalhos. No entender de (Avison e Fitzgerald, 2003), na metodologia ETHICS o desenvolvimento de sistemas baseadas em computadores é vista como um processo de mudanças e por isso envolve conflitos de interesse entre os participantes do processo

O ISAC busca identificar as causas fundamentais dos problemas dos utilizadores. Esta metodologia começa num estágio inicial que muitas metodologias não consideram e não assume que o DSI é necessário para resolver problemas. Se a necessidade do SI não for identificada então o problema termina. A necessidade é estabelecida somente se é visto que o sistema de informação beneficiará as pessoas em seus trabalhos (Avison e Fitzgerald, 2003).

#### 4.10 Considerações sobre as metodologias Desenvolvimento de Sistemas de Informação

Ao desenvolver qualquer projecto de SI, o primeiro grande desafio que os analistas enfrentam é seleccionar qual metodologia a utilizar. Segundo (Denis *et al.*, 2005) escolher uma metodologia não é uma tarefa fácil, porque não há uma metodologia que é sempre melhor. No entanto há muitas organizações que têm as suas normas e políticas que orientam na escolha das metodologias.

A escolha de uma metodologia de ser feita de acordo com a experiencia do analista e da sua equipa de desenvolvimento, levando em consideração as técnicas e ferramentas utilizadas por estes. Conforme (Denis *et al.*, 2005) na selecção de uma metodologia para o desenvolvimento de sistemas deve-se levar em consideração os seguintes aspectos: a necessidade do requisito dos utilizadores, a familiaridade com as tecnologias, a complexidade do sistema, o prazo.

Por seu lado (Avison e Fitzgerald, 2003) reconhecem também que a comparação entre metodologias de DSI não é uma tarefa fácil, e apresentam alguns elementos que permitem uma comparação entre elas, que são:

- ✓ A Filosofia: cada metodologia é desenvolvida de acordo com um paradigma de DSI, têm um dado objectivo e um alvo, ou seja, tem a sua filosofia de desenvolvimento;
- ✓ O Modelo: cada metodologia tem um modelo que é uma abstracção e representação dos factores importantes do SI e da organização;
- ✓ Âmbito: que permite a identificação do ciclo de vida do desenvolvimento que abrange a metodologia, e uma análise detalhada de cada fase;
- ✓ Técnicas e ferramentas: certas metodologias usam técnicas e ferramenta que a diferencia de outras;
- ✓ Saídas: é importante saber o que a metodologia vai produzir em termos de etapas, e o produto final;
- ✓ Prática: saber quem são os participantes da metodologia, a base dos utilizadores;
- ✓ Produto: quantidade de especificação e documentação produzida;
- ✓ A velocidade com que o produto pode ser desenvolvido;
- ✓ O potencial para a modificação dos requisitos dos utilizadores;

## 5 Consideração sobre o primeiro capítulo

Neste capítulo falou-se do papel importante que SI/TI vem assumindo dentro das organizações, e porque cada vez mais as organizações recorrem a eles para melhorar o seu desempenho. Também destacou-se a necessidade de alinhar o SI com a estratégia do negócio implementada pela organização recorrendo ao PESI, pois só assim se consegue obter vantagens recorrendo os SI/TI.

Seguidamente apresentou-se um conjunto de problemas que surgem ao longo do processo de DSI, e a urgente necessidade de controlar esses problemas, por isso apresentou-se alguns modelos de DSI, e algumas metodologias para fazer o controlo dos desenvolvimentos de sistemas de qualidades.

Assim sendo, apresentou-se o primeiro modelo de DSI designado de modelo cascata, que está na origem de muitas metodologias de DSI, e outros modelo que são: o modelo incremental e o modelo espiral. Também falou-se das metodologias de DSI, onde destacou-se as cujo foco são os processos, outras cujo foco são os dados, bem com outras orientadas a objectos, as do desenvolvimento ágil, orientadas a pessoas, e por fim as orientadas á problemas.

## Capítulo 2: Desenvolvimento da aplicação SIRE

---

### 1 Apresentação e objectivo da aplicação

O objectivo final deste trabalho visa a construção de uma aplicação, aplicando uma metodologia de DSI no processo construção do mesmo, no caso a metodologia RUP. A aplicação designada de SIRE (Sistema de Informação de Recursos humano e Estatística) foi desenvolvida para a Delegação do Ministério da Educação e Desporto do concelho da Praia, e abranger dois gabinetes da referida instituição, são eles: o gabinete dos recursos humanos e o gabinete estatística/planeamento.

Com essa aplicação os seus futuros utilizadores terão mais eficiência e melhor produtividade na gestão das suas tarefas diárias, ajudando-lhes a garantir melhor qualidade na prestação de serviços e consequentemente ajudará a manter uma boa imagem da instituição em causa.

#### 1.1 Enquadramento da Instituição

A Delegação do Ministério da Educação e Desporto do concelho da Praia é uma instituição pública que oferece serviços descentralizados do Ministério da Educação e Desporto, nos concelhos da Praia e da Ribeira Grande de Santiago. A mesma presta serviços de orientação,



coordenação e apoio aos estabelecimentos de ensino não superior, e assume as seguintes responsabilidades:

- ✓ Contribuir para a definição e materialização da política educativa;
- ✓ Assegurar a coordenação e articulação dos vários níveis de ensino não superior, de acordo com as orientações definidas a nível central, promovendo a execução da respectiva política educativa;
- ✓ Desenvolver as acções necessárias á condução do processo de ingresso no ensino superior, em articulação com o serviço central respectivo;
- ✓ Colaborar com os órgãos e serviços do ministério nas actividades da ciência e tecnologia e de controlo pedagógico, administrativo e disciplinar;
- ✓ Coordenar e assegurar o funcionamento das instituições do ensino público;
- ✓ Apoiar a formação em serviço e permanente do pessoal docente e não docente;
- ✓ Distribuir o material e equipamento educativo e zelar pela manutenção e conservação dos mesmos;
- ✓ Garantir o normal funcionamento das escolas que funcionam no âmbito do sistema nacional de educação, em articulação com as direcções dos estabelecimentos do ensino, as entidades locais e a comunidade;
- ✓ Coordenar a elaboração e actualização do cadastro dos equipamentos educativos.

A Instituição encontra-se estruturada em vários gabinetes que se interligam e formam um todo, como se pode ver na estrutura orgânica da mesma, apresentada na figura13:

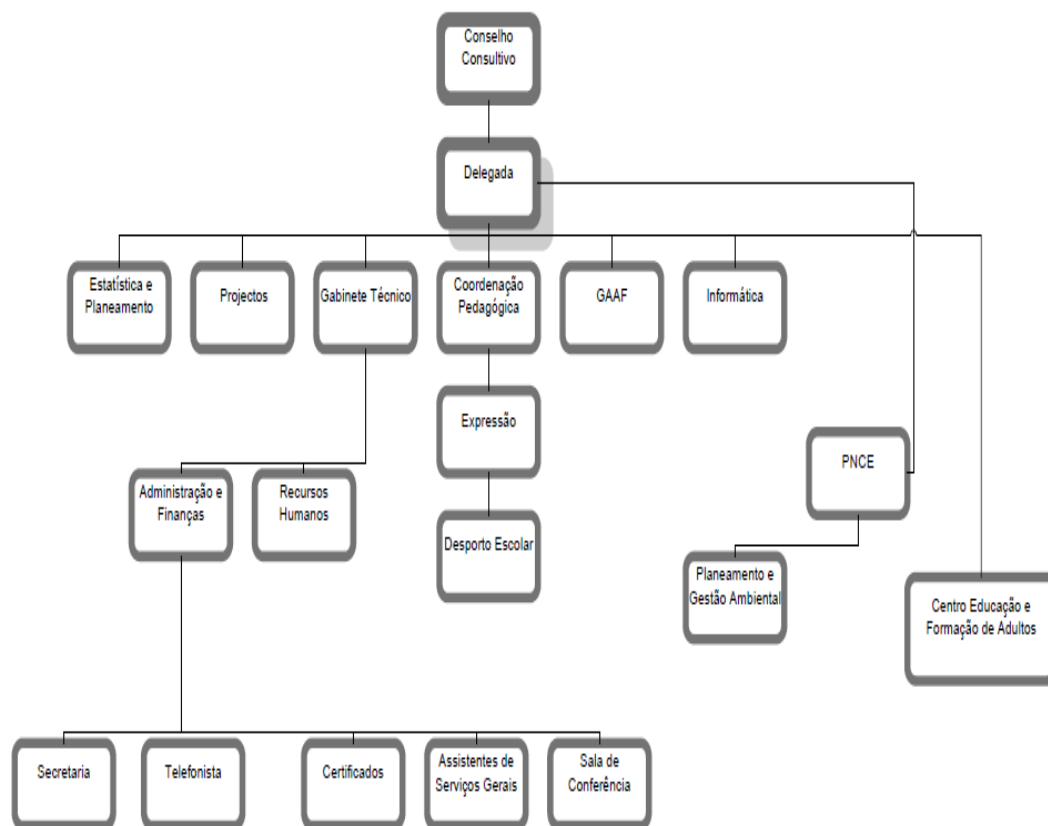


Figura 13: Estrutura Orgânica de Delegação do MED do concelho da Praia

## 2 Ciclo de vida do desenvolvimento da aplicação SIRE

O desenvolvimento da aplicação foi baseado na abordagem orientada a objecto, e a escolha dessa justifica-se pelo facto de algumas das várias vantagens que a orientação a objecto tem vindo proporcionar no desenvolvimento de sistema, como exemplo o factor da herança que permite a reutilização de códigos, e o encapsulamento que “esconde” as propriedades dos objectos garantindo com que os códigos sejam mais robustos e com maior segurança, também a facilidade de introduzir mudanças nas classes.

Foi assim utilizada a metodologias RUP, onde foram abordadas as fases e iterações correspondentes ao ciclo de desenvolvimento de sistema RUP, bem como os macros de cada fase e algumas das características importante desta metodologia.

Antes de começar a construir o sistema, fez-se um plano para cada fase de desenvolvimento, e um conjunto de tarefas afectos a esta, e também alguns macros para cada fase de desenvolvimento, como se pode ver na tabela de planeamento das fases no apêndice i.

## 2.1 Conceção

Nesta primeira fase de desenvolvimento do sistema é apresentada uma visão geral do problema, ou seja, apresenta-se a justificação inicial do projecto através da determinação do problema, e é também indicada a viabilidade do projecto, bem como os referidos requisitos iniciais.

### ✓ **Descrição do problema e Justificação do Projecto:**

A Delegação do Ministério da Educação e Desporto do concelho da Praia está fragmentada por diversos gabinetes que trabalham sobre as informações relacionados com os professores, alunos e escolas.

Porém esses gabinetes trabalham essas informações de uma forma isoladas, pois as mesmas se encontram centralizadas nos computadores de cada funcionários não havendo uma base de dados central para a partilha dessa informação, o que consequentemente leva a duplicação e redundância dos dados, e põe em causa a consistência dos mesmos. Também há uma grande dificuldade na manutenção destas informações visto que se encontram descentralizados, e outros em formatos impressos. Sem esquecer a constantes perdas de algumas informações, causadas pelos vírus dos computadores.

Perante estas lacunas surgiu a necessidade de centralizar essas informações num único repositório, de forma a serem compartilhadas pelos funcionários das diversas áreas da instituição e a partir disto obter-se um maior controlo dessas informações. Assim nasceu a aplicação de gestão interna que na sua fase inicial irá beneficiar os gabinetes de recursos humanos e estatística/planeamento.

A aplicação desenvolvida é viável porque trará para aos funcionários da instituição um maior rigor nos seus trabalhos diários e permitirá que estes façam um planeamento das dessas tarefas com maior eficiência, e trará benefícios tais como:

- Possibilita uma melhor gestão sustentado por maior quantidade de informações e com mais eficiência, ou seja, possibilitará projectar dados futuros para posteriores análises, como por exemplo: saber a informação de todos os funcionários que irão se reformar num dado ano, ajudará a traçar metas de quantos novos funcionários terão que contratar;
- Possibilita a troca de informação com maior rapidez, e reduz a perda de tempo;
- Possibilita maior coerência sobre as informações, e menos duplicação de dados, ou seja, haverá uma base de dados central que permitirá a troca de informações mais concisa, eliminando a duplicação de informações;
- Maior entusiasmo no trabalho diários dos utilizadores, já que as informações disponibilizadas se encontram bem estruturados e de fácil utilização;

✓ **Requisitos iniciais do sistema**

As informações básicas sobre o sistema foram levantadas recorrendo a entrevista informal com os futuros utilizadores do sistema, bem como a análise de documentos em formatos impressos e digitais que facultavam informações necessárias para o desenvolvimento do referido sistema de informação. Desta feita pretendem-se que o SIRE tenha as seguintes funcionalidades:

- Identificação dos utilizadores: o sistema deverá ter um mecanismo de identificação dos utilizadores, para isso, o sistema deve ter informações de todo os futuros utilizadores dessa base de dados, igualmente deve conter informações sobre as permissões de cada um para assim atribuir tarefas consoante a permissão do utilizador;

- Permitir a inserção/alteração e desactivação dos dados dos funcionários que trabalham nas escolas ligados com a delegação bem como os funcionários da instituição;
- Permitir a inserção/alteração e desactivação dos dados dos professores que estão ligados á Delegação e inserir também os que estão como coordenadores, gestores, os que estão substituição, em licença ou na secretaria;
- Permitir a inserção/alteração e listar todos os funcionários reformados;
- Permitir a inserção/alteração dos dados dos alunos que estão no ensino básico integrado;
- Permitir que os utilizadores possam fazer consultas de forma a obter dados estatísticos, como por exemplo consultar por todos funcionários de uma dada escola, consultar por todos os professores, saber o quem são os professores coordenadores, os que estão na secretaria, os que são gestores. Também os que estão em licenças bem como os que estão em substituição;
- Permitir aos utilizadores fazerem consultas relativamente aos alunos, como por exemplo a quantidade de alunos por escolas, a quantidade de alunos por ano electivo, quantidades de alunos aprovados e reprovados, entre outras consultas que poderão ser extraídas da base de dados.
- A partir dos requisitos iniciais estabelecidos, foram identificados três categorias dos futuros utilizadores do sistema:
- Administrador do sistema: É o responsável pelo sistema, terá a responsabilidade de garantir a rica evolução do sistema, e é quem controla tudo.
- Recursos humanos: destinada ao seu director, e usufrui de um conjunto de tarefas destinadas ao director dos recursos humanos.

- Estatística e planeamento: destinada aos utilizadores do gabinete de estatística e planeamento, estes usufruem também de um conjunto de actividades direccionadas a eles.

De forma a facilitar a identificação das permissões dos utilizadores, os mesmos serão descritos com letras A, R, E correspondentes aos utilizadores referidos anteriormente (Administrador, Recursos Humanos, Estatística e Planeamento).

## 2.2 Elaboração

Nesta segunda fase é detalhada os requisitos iniciais do sistema, e é apresentada um conjunto de requisitos funcionais, bem como a descrição da arquitectura de software. Assim sendo, levou-se em consideração as informações iniciais do sistema descrito na fase de concepção, e detalhou-se as mesmas através da modelação de sistema, tendo sempre em consideração que o sucesso do sistema depende da sua boa ou má modelação.

Para a modelação do sistema teve-se em consideração de que a metodologia RUP é baseada em UML, por isso foram utilizadas alguns dos diagramas UML (diagrama de Caso de Utilização, diagrama de classe, diagrama de pacotes e diagrama de sequência) para melhor representar o sistema.

Também levou-se em consideração outras características importantes desta metodologia, que é o caso de esta ser centrada em diagrama de casos de utilização, pelo facto desse diagrama permitir capturar os requisitos funcionais do sistema, ou seja, as necessidades dos utilizadores são traduzidos em modelo de casos de utilização. E a partir deste, o processo de desenvolvimento segue um fluxo, em que os diagramas de casos de utilização são especificados, desenhados e implementados. Por isso foi construindo o diagrama de caso de utilização para melhor entender os requisitos do sistema como ilustrada na figura14.

### ✓ Requisitos funcionais do sistema:

Os requisitos funcionais do sistema estão representados na figura14, onde encontram-se representados no diagrama três actores que interagem com o sistema que são: administrador do sistema, director de recursos humanos, e o pessoal da área de estatística/planeamento, bem como as suas respectivas tarefas, ou seja, cada caso de utilização representa um requisito do sistema.

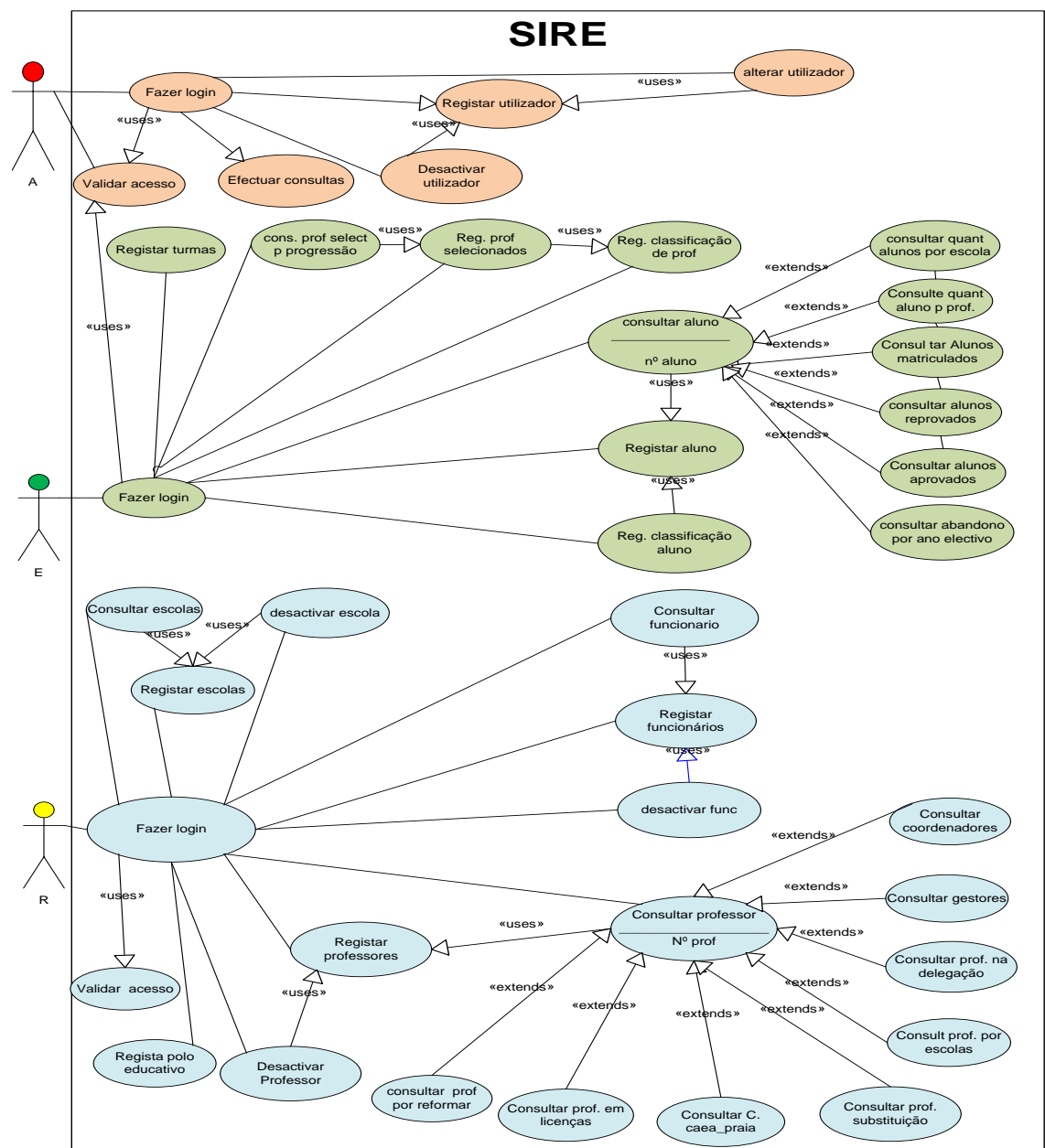


Figura 14: Diagrama Caso de utilização

De seguida é apresentado o diagrama de sequência que mostra a interacção entre o director dos Recursos Humanos e as classes que lhe são afectos, representada na figura15.

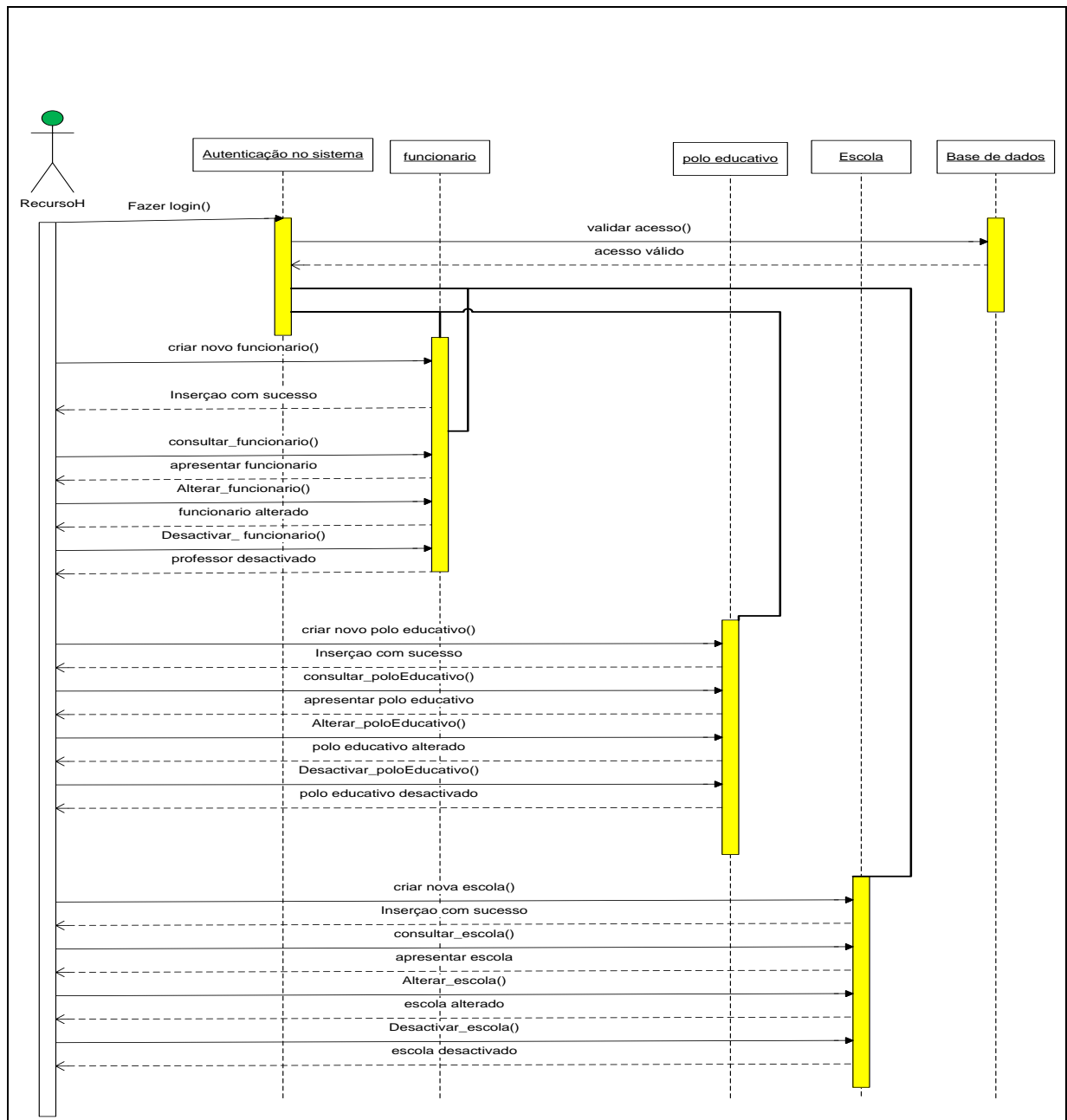


Figura 15: Diagrama de sequência



A partir do diagrama de casos de utilização foi analisada e construída a arquitectura lógica do sistema representada através dos diagramas de classes, como se pode ser na figura16.

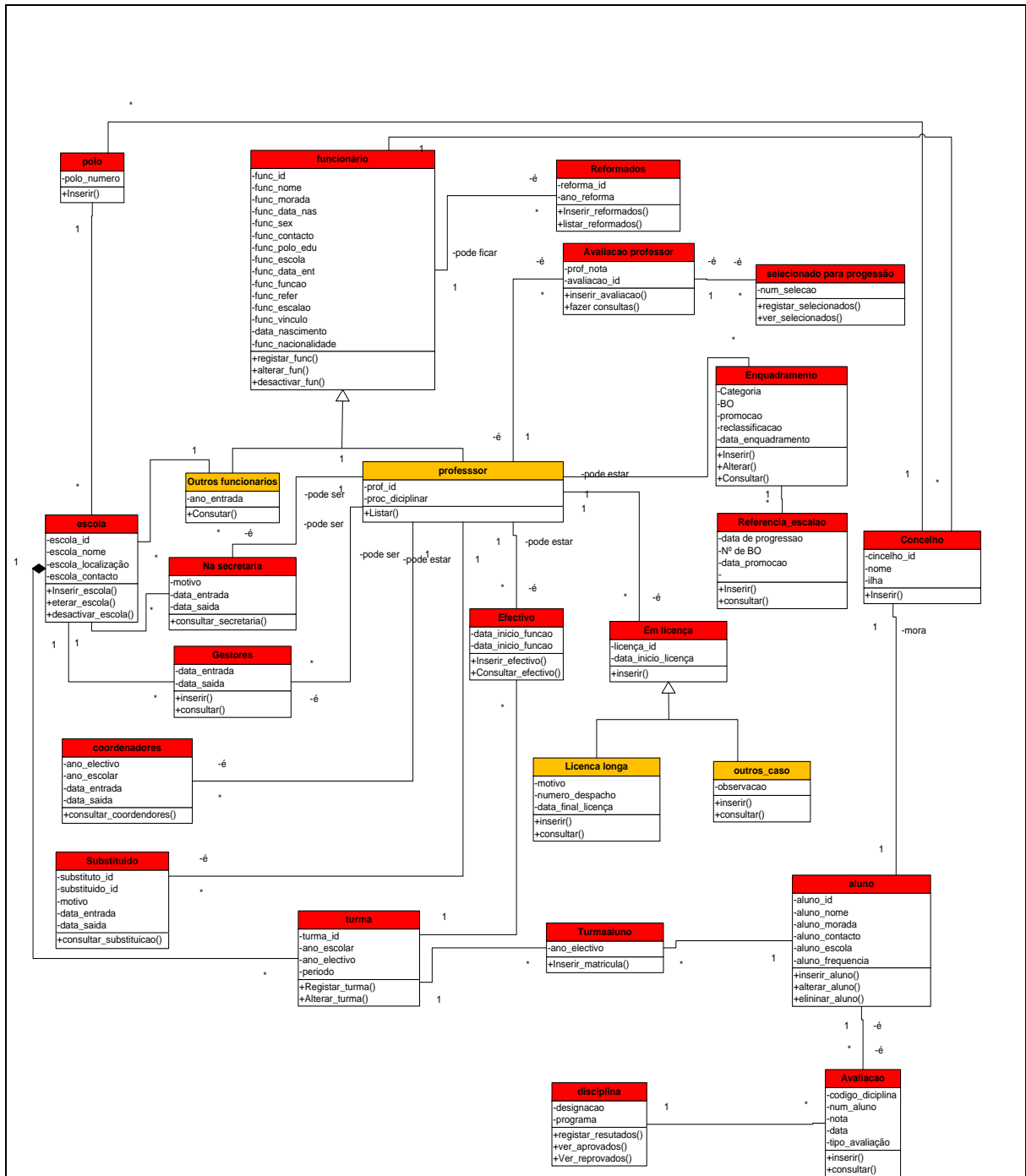


Figura 16: Diagrama de classe

A partir do diagrama de classe obteve-se o modelo entidade/relacionamento, que por seu lado representa a forma como a informação serão inseridas na base de dados, representadas na figura 17.

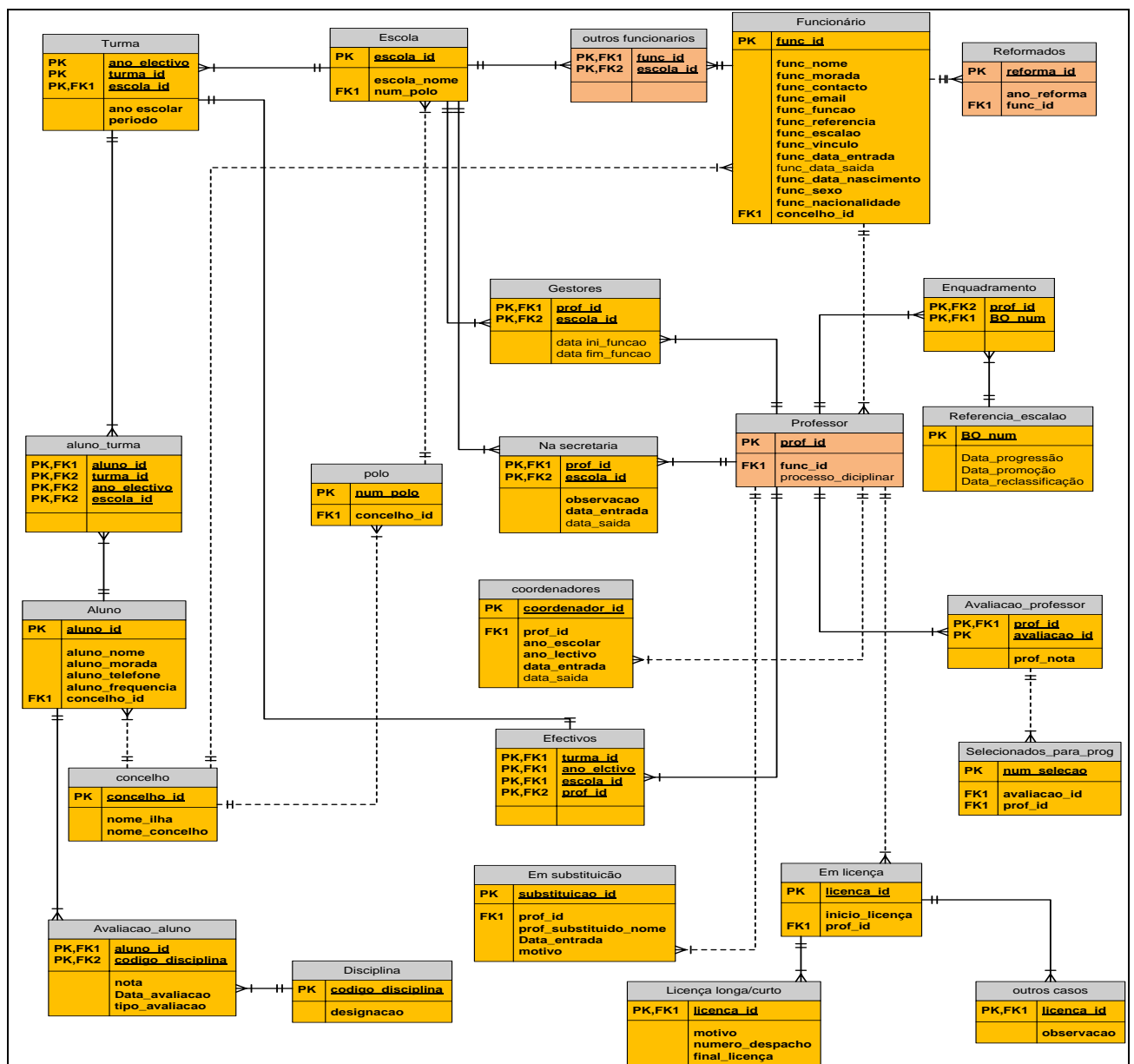


Figura 17: Diagrama Entidade/ Relação

A metodologia RUP é centrada na arquitectura, o que permite ter uma visão geral dos subsistemas e como estes se relacionam, por isso o sistema foi dividido em subsistemas,

construídos a partir do diagrama de classe apresentado anteriormente, para desta forma poder obter uma melhor gestão da aplicação. Desta feita foi construído um diagrama de pacotes que ilustra a divisão do sistema em módulo, que são:

- ✓ Subsistema “Controlo de acesso” que é responsável pelo controlo a acesso á aplicação;
- ✓ Subsistema “Gestão de aluno” que permitirá fazer toda a gestão dos alunos;
- ✓ Subsistema “Gestão de escolas” que permitirá fazer a gestão escolas;
- ✓ Subsistema “Gestão de professores” que permitirá fazer toda a gestão dos professores;
- ✓ Subsistema “Gestão de funcionários” que permitirá fazer toda a gestão dos funcionários;

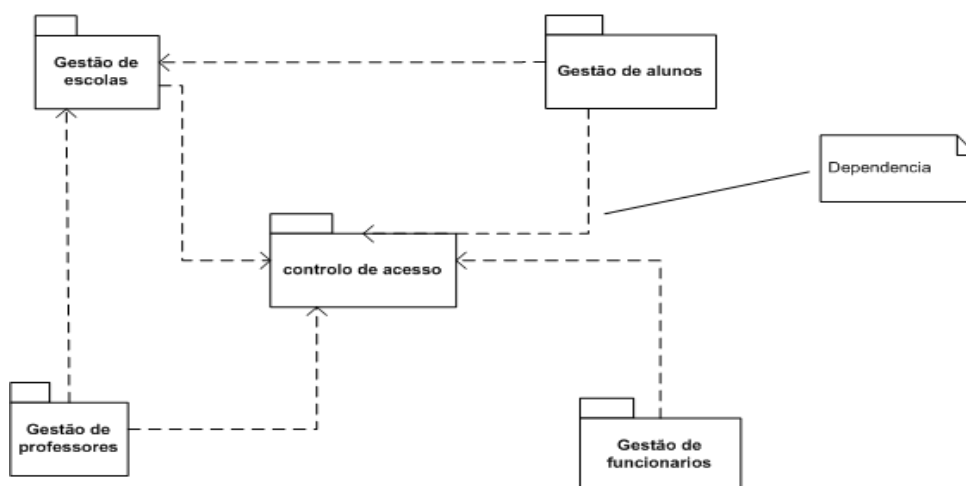


Figura 18: Subsistema

### ✓ **Arquitectura do sistema**

O funcionamento da de aplicação baseia-se na arquitectura cliente/servidor, com 3 camadas, que serão descritas de seguida:

Primeira camada é caracterizada pelo cliente Web, com a capacidade de apresentação das páginas Web, sobre um protocolo http, em que o utilizador requisita um ficheiro que se encontra num computador remoto através de um endereço (URL). Este pedido é efectuado através de um navegador Web.

A segunda camada corresponde a camada de apresentação, onde o Servidor de Web compreende o pedido do cliente e devolve o respectivo ficheiro.

A terceira camada é a camada de dados no qual reside todas as informações para o funcionamento da aplicação.

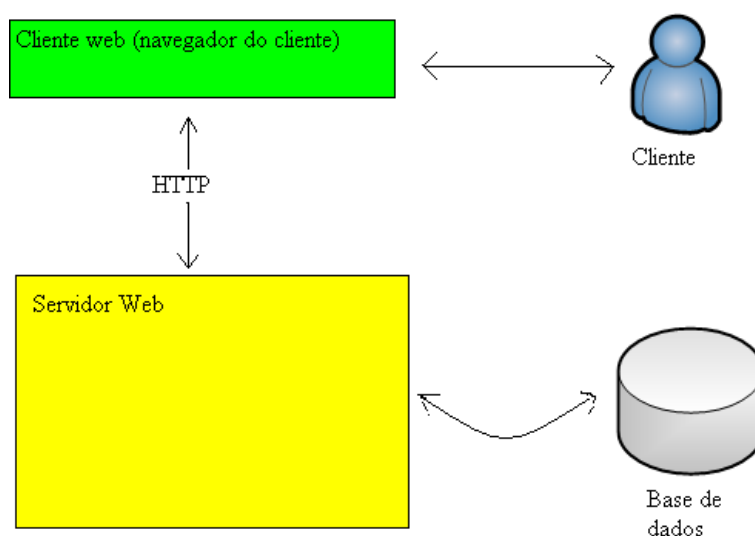


Figura 19: Arquitectura do sistema

## 2.3 Construção

A fase de construção tem como o objectivo a construção e implementação da aplicação. A implementação do sistema foi baseada em incrementos, onde as necessidades prioritárias foram construídas primeiro.

Desta feita para que se avançar com a programação fez-se uma análise dos requisitos dos utilizadores e atribuiu-lhes prioridades, para isso construiu-se uma tabela de prioridades dos requisitos para auxiliar na construção do sistema, como se pode ver no apêndice ii, onde atribuiu-se maior prioridades as necessidades do requisitos do director dos recursos humanos em relação aos requisitos de do restantes utilizadores, dado que as urgências das necessidades do mesmo.

No processo de construção do sistema foi levado em consideração os diagramas UML construídos anteriormente, principalmente o diagrama de classes, onde para cada classe foi construída uma classe de implementação, formam também codificadas as associações existentes entre as classes. E o sistema foi codificado em linguagem PHP. Seguidamente é apresentadas e descritas algumas das interfaces funcionais mais importantes do sistema.



Figura 20: Página de autenticação do sistema

A figura 20 representa a pagina inicial do sistema, em que para ter acesso aos menus, o utilizador terá que fazer a autenticação mediante um *username* e um *password*, caso o *username* ou *password* estiverem incorrecto receberá automaticamente uma mensagem de erro. Caso contrário, o utilizador receberá os seus menus consoante o seu perfil, que pode ser administrador, recursos humanos e estatística. Seguidamente será ilustrada os menus mais importantes para recursos humanos.



Figura 21: Ambiente gráfico do Director dos recursos humano

A figura 21 representa o ambiente gráfico do director dos recursos humanos, para uma melhor gestão os menus se encontram separados por diferentes categorias que são: a escola, os funcionários, os professores e reformados. Cada categoria dos menus se encontra subdivididos em pequenos submenus, que são pequenas tarefas que podem ser realizadas em cada categoria, como se pode ver na figura 22.

**SIRE**  
DIRECTOR DE RECURSOS HUMANOS

**Menu Recursos Humanos**

- Home
- Escola
- Funcionarios
  - Adicionar
  - Consultar
  - Alterar
  - Desativar
- Professores
- Reformados

**Formulário de Funcionário:**

Nome:

Data Nascimento:

Nacionalidade:

Morada:

Concelho:

Telefone:

Email:

Sexo:

Funcao:

Referência/Escalao/Vinculo: Referência:  Escalão:  Vinculo:

Data Entrada:

Data saída:

Nome a Escola:

Figura 22: Submenus da categoria funcionários

Uma das tarefas muito exigido pelo utilizador dos recursos humanos foi a opção obter consultas a base de dados, onde a partir dessas informações poderá projectar dados futuros que lhe ajudarão a melhorar o seu trabalho diário, algumas dessas consultas estão ilustradas nas figuras 23, 24.

**SIRE**  
DIRECTOR DE RECURSOS HUMANOS

**Menu Recursos Humanos**

- Home
- Escola
- Funcionarios
  - Adicionar
  - Consultar
  - Alterar
  - Desativar
- Professores
- Reformados

**Todos os Funcionários**

Nome:	Morada	Data Nascimento	Sexo	Nacionalidade	Telefone	Email	Funcao	Local de Trabalho
Marley Goncalves	ponta dagua	1996-10-11	feminino	caboverdiana	2641256	mc@hotmail.com	Ajudante de Serviço Gerais	vila nova
Alberto Moniz	Sao filipe	1976-10-08	masculino	Portuguesa	264	albertomz@hotmail.co	Condutor	vila nova
Ana Correia	Fazenda	2001-10-10	feminino	caboverdiana	2623145	ana@hotmail.com	Ajudante de Serviço Gerais	lem ferreira

Figura 23: Consulta a sobre funcionário

SIRE

DIRECTOR DE RECURSOS HUMANOS

Menu Recursos Humanos

Home

Escola

Funcionarios

Professores

Adicionar

Consultar

Alterar

Inserir Categoria

Desativar

Reformados

PROFESSORES EM SUBSTITUIÇÃO

Nome do Substituido:	Nome do substituto	Data inicio	Data Fim	Motivo
Marcos de Pina	Jailson Fortes	2011-10-04	2012-01-04	Internacao
Jussara Mendes Perreira	Maria Tavares	2011-10-04	2012-01-04	saude
Marcos de Pina	Joaninha lima	2008-10-10	2010-03-12	problemas de saude

Figura 24: Consultas professores em Substituição

## 2.4 Transição:

Essa última fase tem como objectivo a distribuição do produto final ao cliente. Sendo assim, o produto final foi entregue ao responsável do gabinete de informática da instituição baseada em incrementos e mediante interacção dos futuros utilizadores. Foi também entregue a documentação do processo de DSI para posteriores evoluções e manutenção do referido sistema.

## 3 Considerações sobre o segundo capítulo

Neste capítulo fez-se a construção de uma aplicação baseado na metodologia RUP, onde foi aplicado algumas das práticas consideradas importantes nesta metodologia. No entanto, uma das partes mais importante da construção do sistema, foi a modelação do mesmo, visto a modelação é de extrema importância para o sucesso do sistema, por isso procurou-se concentrar nesta fase crítica, despendendo horas para que o sistema fosse bem modelado.



Entretanto, a aplicação da metodologia RUP no processo construção desse sistema ajudou a compreender melhor as necessidades do novo sistema através da utilização do diagrama de caso de utilização, e sobretudo o ciclo de DSI da mesma possibilitou acelerar a construção do sistema, e ajudou construir um novo sistemas com rigor e qualidade, já que através disso o sistema ficou bem detalhado e de fácil compreensão.

## Conclusão:

Como se pode constatar durante o trabalho, os SI/TI têm vindo a assumir um estatuto importante dentro das organizações, por isso conclui-se que estes servem como a ponte entre a organização e a qualidade dos seus serviços, querendo dizer que actualmente as organizações vêm-se forçadas a adaptarem os seus sistemas consoantes as necessidades, sobre pena de estes ficarem obsoletos e de pouca utilidade.

No entanto, no decorrer do trabalho também pode se constatar que os SI/TI por si só não garantem as vantagens competitivas que tanto as organizações procuram, se estes não forem alinhados com as estratégias da organização, concluindo que mesmo que as organizações tenham as mais sofisticadas TI, se estes não formem alinhadas com a estratégias organizacional podem não trazer vantagens para a mesma, portanto surge a necessidade de se fazer um PESI, de forma a satisfazer a estratégia organizacional e apresentar propostas de TI que podem ajudar a melhorar o desempenho destas. Também o PESI é um processo que deve envolver todos os intervenientes da organização.

Constatou-se que diversos problemas podem surgir durante a realização no processo de DSI, e assim concluiu-se que as metodologias de DSI são uma via para resolução desses problemas, e quando aplicado de forma correcta num determinado contexto, esta poderá se revelar de muita utilidade.

No entanto, existem diversas metodologias de DSI, que têm o propósito de melhorar a qualidades dos SI desenvolvidos, umas que dão importância a processos, outras a dados, outras orientadas a objectos, entre outras metodologias utilizadas no processo DSI. Todavia, conclui-se que não se pode considerar uma metodologia melhor que a outra, pois cada uma apresenta a sua filosofia de desenvolvimento, têm as suas limitações e pode ser aplicado num dado contexto.

Assim sendo aplicou-se a metodologia RUP no desenvolvimento da aplicação *Web* para a Delegação do Ministério da Educação e Desporto do concelho da Praia. Essa metodologia foi

de muita utilidade e trouxe maior eficiência no decorrer desse processo, pois possibilitou a construção de aplicação mais iterativa, dado que esta atribui grande importância aos requisitos do sistema, e através desta foi possível o desenvolvimento desse sistema perante uma estrutura bastante rigorosa que garanta a qualidade do mesmo quando implementado, ou seja, através das fases do ciclo de vida da metodologia RUP foi possível fazer uma análise profundo do sistema e com maiores detalhes, o que garante a qualidade do sistema.

## Bibliografias

Agile Manifesto: History The Agile Manifesto <<http://www.agilemanifesto.org/history.html>>, Consultado em Setembro 2011.

Agile Manifesto: Home Page <<http://www.agilemanifesto.org/>>, Consultado em Setembro 2011.

Agile Manifesto: Principles behind the Agile Manifesto <<http://www.agilemanifesto.org/principles.html>>, Consultado em Setembro 2011.

Almeida A. (2002), *Qualidade no software*, Universidade Aberta.

Avison D., Fitzgerald G. (2003), *Information Systems Development: Methodologies, techniques and Tools (Third Edition)*, McGraw-Hill Education.

Beynon-Davies P. (2002), *Information Systems: An Introduction to Informatics in organizations*, Palgrave.

Da Luz Carlos, *Desenvolvimento de Sistemas de informação*, Universidade Jean Piaget, Praia 2011.

Dennis A., Wixom Barbara, Tegarden D. (2005), *Systems Analysis and Design with UML Version 2.0: An object-Oriented Approach, Second Edition*, John Wiley & Sons.

Dall'Oglio P. (2009), *Programando com orientação a objetos (2ª Edição)*, Novalac.

Drifa S. (2003), *Merise: concepts et mise en oeuvre Ressources Informatiques*, ENI.

Extreme Programming: The Values of Extreme Programming <<http://www.extremeprogramming.org/values.html>>, Consultado em Setembro 2011.

Hoffer J., George J., Valacich J. (2002), *Systems analysis & design*, Prentice Hall.

IBM Software: Rational Agile Development <<http://www-01.ibm.com/software/rational/agile/>>, Consultado em Setembro 2011.

Isaías P. (2001), *Análise de Sistema de Informação*, Universidade Aberta.

- Kroll P., Kruchten P. (2003), *The Rational Unified Process made easy: a practitioner's guide to the RUP*, Pearson Education.
- Kruchten P. (2004), *The Rational Unified Process: An introduction (Third edition)*, Pearson Education.
- Lopes F., Morais M., Carvalho A. (2005) *Desenvolvimento de sistema de informação*, FCA.
- Lejk M., Deeks D. (2002), *An Introduction to Systems Analysis Techniques*, Pearson Education.
- Laudon K., Laudon J. (2006), *Management Information Systems*, Pearson Education.
- Laudon K., Laudon J. (1995), *Essentials of Management Information Systems*, Prentice Hall.
- Moraes R., Chiossi T. (2006), *Especificação de sistemas de software utilizando análise e projecto estruturados*, UNICAMP.
- Nunes Mauro, O'Neill Henrique, (2004) *Fundamentos de UML (3ª Edição Actualizada)*, FCA.
- Rodrigues L. (2002), *Arquitecturas dos sistemas de informação*, FCA.
- Ramos R. (2006), *Treinamento prático Em UML*, Digerati Books.
- Silva A., Videira C. (2005), *UML metodologias e ferramentas case (volume1)*, Centro atlântica.
- Silva A., Videira C. (2008), *UML metodologias e ferramentas case (volume 2)*, Centro atlântica.
- Serrão C., Marques J. (2007), *Programando com PHP5*, FCA.
- Serrano A., Caldira M., Guerreiro A. (2004), *Gestão de sistemas e tecnologias de sistemas de informação*, Editora FCA, Lisboa.
- Sommerville I. (2003), *Engenharia de Software (6ª Edição)*, Addison Wesley.
- Varajão J. (1998), *Arquitectura da gestão de sistemas de informação*, FCA.

Xprogramming An Agile Software Development: What is Extreme Programming,  
<<http://www.xprogramming.com/what-is-extreme-programming/>>, Consultado em Setembro  
2011.

# Apêndices

i. Planeamento das fases de desenvolvimento da aplicação SIRE

✓ Conceção	<ul style="list-style-type: none"><li>▪ Contacto com a Instituição/Compreensão do problema;</li><li>▪ Construção do Cronograma de execução do projecto;</li><li>▪ Levantamento dos requisitos iniciais do sistema;</li></ul> <p>Marco: objectivos do ciclo de vida do sistema;</p>
✓ Elaboração	<ul style="list-style-type: none"><li>▪ Definição do sistema;</li><li>▪ Construção/descrição do diagrama de caso de utilização, e periodização de caso de utilização/Construção de diagrama de classe;</li><li>▪ Esquema de base de dados/ Modelo de arquitectura do sistema;</li></ul> <p>Marco: Arquitectura de ciclo de vida;</p>
✓ Construção	<ul style="list-style-type: none"><li>▪ Codificação do sistema;</li></ul> <p>Marco: Capacidade operacional do sistema;</p>
✓ Transição	<ul style="list-style-type: none"><li>▪ Plano de implementação</li><li>▪ Formação dos futuros utilizadores;</li></ul> <p>Marco: Entrega do produto ao utilizador;</p>



## ii. Prioridade de caso de utilização

	Mais prioritários	Menos prioritários
Fazer Login/Validar acesso	X	
Registar/Alterar/Desactivar/ Conjunto de Consultas afectos aos Funcionários	X	
Registar/Alterar/Desactivar/ Conjunto de Consultas afectos aos Professores	X	
Registar/Alterar/Desactivar/ Conjunto de Consultas afectos as Escolas		X
Registar/Alterar/Desactivar/ Conjunto de Consultas afectos aos Alunos		X
Registar/Alterar / Conjunto de Consultas afectos aos Reformados	X	
Registar/consultar/Alterar avaliação de alunos e professores		X